

21.158/1A/4/05



MILIK PERPUSTAKAAN  
INSTITUT TEKNOLOGI  
SEPULUH - NOPEMBER

# PERANCANGAN DAN PEMBUATAN PERANGKAT LUNAK UNTUK MENGLASIFIKASIKAN CITRA KOROSI MENGGUNAKAN METODE ANALISIS TEKSTUR

## TUGAS AKHIR



RSIF  
005-1  
Ahm  
P-1  
2000

Oleh :

**TOHARI AHMAD**  
2695 100 050

PERPUSTAKAAN ITS	
Tgl. Terima	16-7-2003
Terima Dari	H/
No. Agenda Prp.	210413

**JURUSAN TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INDUSTRI  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA  
2000**

# **PERANCANGAN DAN PEMBUATAN PERANGKAT LUNAK UNTUK MENGLASIFIKASIKAN CITRA KOROSI MENGGUNAKAN METODE ANALISIS TEKSTUR**

## **TUGAS AKHIR**

**Diajukan Guna Memenuhi Sebagian Persyaratan Untuk Memperoleh**

**Gelar Sarjana Komputer**

**Pada**

**Jurusan Teknik Informatika**

**Fakultas Teknologi Industri**

**Institut Teknologi Sepuluh Nopember**

**Surabaya**

**Mengetahui / Menyetujui**

**Dosen Pembimbing I**



**Ir. ESTHER HANAYA, M.Sc**  
**Nip. 130 816 212**

**Dosen Pembimbing II**



**RULLY SOELAIMAN, S. Kom**  
**Nip. 132 085 802**

**Surabaya**

**Agustus 2000**





## ABSTRAK



## ABSTRAK

Korosi muncul pada hampir semua material yang ada dan berada pada lingkungan yang bervariasi. Dua morfologi dasar korosi adalah *pit* dan *crack*. Kedua morfologi tersebut dapat berada pada material yang sama dan mempunyai penampakan yang berbeda.

Tujuan tugas akhir ini adalah untuk mengklasifikan kedua morfologi tersebut. Permasalahan yang muncul di sini adalah memisahkan morfologi korosi dari *background*-nya. Korosi mempunyai bentuk yang bervariasi, sehingga pemisahan morfologi korosi dengan cara segmentasi kurang menunjukkan hasil yang baik.

Dalam permasalahan ini digunakan metode yang tidak memerlukan segmentasi, yaitu dekomposisi wavelet, baik wavelet standar maupun wavelet paket, untuk mendapatkan *feature* sebagai data masukan jaringan syaraf. Metode jaringan syaraf yang dipakai di sini adalah *Learning Vector Quantization (LVQ)* dengan beberapa variasinya, yaitu *LVQ1*, *LVQ2.1* dan *OLVQ1*.





## KATA PENGANTAR



## KATA PENGANTAR

Alhamdulillah puja dan puji syukur penulis panjatkan ke hadirat Allah SWT, karena atas izin dan berkah-Nya penulis telah menyelesaikan tugas akhir dengan judul :

### **PERANCANGAN DAN PEMBUATAN PERANGKAT LUNAK UNTUK MENGKLASIFIKASIKAN CITRA KOROSI MENGGUNAKAN METODE ANALISIS TEKSTUR**

Tugas akhir ini merupakan syarat akademis untuk menyelesaikan pendidikan di Jurusan Teknik Informatika, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember Surabaya.

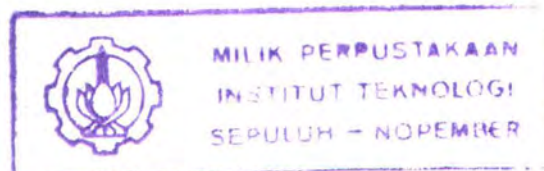
Dengan selesainya tugas akhir ini, penulis ingin memberikan penghargaan sebesar-besarnya kepada semua pihak yang telah dengan ikhlas memberikan bantuannya dengan mengucapkan terima kasih yang setulus-tulusnya dan penghargaan sebesar-besarnya serta doa semoga Allah SWT membalas bantuan mereka. Untuk itu perkenankanlah pada kesempatan ini penulis menyampaikan rasa terima kasih yang sebesar-besarnya kepada :

1. Ir. Esther Hanaya, M.Sc., selaku dosen pembimbing I yang telah memberikan bimbingan, petunjuk dan pengarahan kepada penulis dalam menyelesaikan tugas akhir ini.
2. Rully Soelaiman, S.Kom., selaku dosen pembimbing II yang telah memberikan bimbingan, petunjuk dan pengarahan kepada penulis dalam menyelesaikan tugas akhir ini.
3. Seluruh staf dan dosen Teknik Informatika ITS.



## DAFTAR ISI

	Halaman
ABSTRAK.....	ii
KATA PENGANTAR.....	iii
DAFTAR ISI.....	v
DAFTAR GAMBAR.....	viii
DAFTAR TABEL.....	x
 BAB I PENDAHULUAN.....	 1
1.1 Latar Belakang.....	1
1.2 Perumusan Masalah.....	2
1.3 Batasan Masalah.....	4
1.4 Tujuan Tugas Akhir.....	4
1.5 Manfaat Tugas Akhir.....	4
1.6 Metodologi.....	4
1.7 Sistematika Penulisan.....	5
 BAB II DASAR TEORI WAVELET DAN MULTIRESOLUSI.....	 7
2.1 Wavelet.....	7
2.2 Multiresolusi .....	12
2.2.1 Analisis Wavelet.....	12
2.2.1.1 Transformasi Wavelet Kontinu.....	13
2.2.1.2 Transformasi Wavelet Diskret.....	13
2.2.2 Analisis Multiresolusi.....	14
2.3 Analisis Citra Berdasarkan Wavelet.....	17
2.4 Analisis Tekstur Berdasarkan Wavelet.....	19
 BAB III KLASIFIKASI POLA.....	 24
3.1 Definsisi .....	24
3.2 Pengklasifikasi/ <i>Classifier</i> .....	25





3.3	<i>Learning Vector Quantization</i> .....	29
3.3.1	Arsitektur.....	29
3.3.2	Algoritma.....	30
3.4	Performansi.....	31
BAB IV	PERANCANGAN DAN PEMBUATAN PERANGKAT LUNAK .....	32
4.1	Pendahuluan.....	32
4.2	Perancangan Perangkat Lunak dan Tujuannya.....	32
4.2.1	Perancangan Arsitektur Jaringan Syaraf.....	32
4.2.2	Perancangan Data.....	33
4.2.2.1	Data Masukan.....	34
4.2.2.2	Data Proses.....	34
4.2.2.3	Data Keluaran.....	35
4.2.3	Perancangan Proses.....	35
4.2.4	Hirarki Modul.....	46
4.3	Pembuatan Perangkat Lunak .....	47
4.3.1	Implementasi Struktur Data.....	47
4.3.2	Implementasi Proses.....	49
4.3.2.1	Proses Inisialisasi.....	49
4.3.2.2	Proses Pelatihan.....	50
4.3.2.3	Proses Klasifikasi.....	53
4.4	Tampilan Antarmuka.....	53
4.4.1	Insialisasi.....	54
4.4.2	Pelatihan.....	54
4.4.3	Klasifikasi.....	54
BAB V	UJI COBA DAN PEMBAHASANNYA.....	58
5.1	Uji Coba dan Analisis Hasil Klasifikasi.....	58
5.1.1	Pengujian Data Menggunakan 14 Citra Pelatihan.....	60
5.1.2	Pengujian Data Menggunakan 28 Citra Pelatihan.....	60
5.1.3	Pengujian Data Menggunakan 3 Codebook Terbaik.....	61
5.1.4	Pengujian Data Citra yang Dicerminkan Garis Horizontal....	61



5.1.5 Pengujian Data Citra yang Berisi Korosi Kelas Pit dan Crack.....	62
5.2 Uji Coba dan Analisis Kecepatan Proses Sistem.....	65
BAB VI PENUTUP.....	70
6.1 Kesimpulan.....	70
6.2 Saran.....	71
DAFTAR PUSTAKA.....	72
LAMPIRAN	

## DAFTAR GAMBAR

	Halaman
Gambar 2.1 Proses dekomposisi.....	18
Gambar 2.2 Wavelet standar.....	23
Gambar 2.3 Wavelet paket.....	23
Gambar 3.1 Pendekatan jaringan syaraf.....	27
Gambar 3.2 Arsitektur jaringan.....	29
Gambar 4.1 Arsitektur jaringan syaraf.....	33
Gambar 4.2 Dekomposisi 3 feature.....	36
Gambar 4.3 Dekomposisi 5 feature.....	36
Gambar 4.4 Dekomposisi 4 feature.....	36
Gambar 4.5 Dekomposisi 8 feature.....	36
Gambar 4.6 DFD level 0.....	37
Gambar 4.7 DFD level 1.....	38
Gambar 4.8 Diagram alir proses.....	39
Gambar 4.9 Diagram alir proses inisialisasi.....	40
Gambar 4.10 Diagram alir proses citra.....	41
Gambar 4.11 Diagram alir wavelet standar.....	42
Gambar 4.12 Diagram alir wavelet paket.....	43
Gambar 4.13 Diagram alir proses pelatihan.....	44
Gambar 4.14 Diagram alir proses klasifikasi.....	45
Gambar 4.15 Hirarki modul sistem klasifikasi.....	46
Gambar 4.16 Tampilan inisialisasi.....	55
Gambar 4.17 Tampilan pelatihan.....	56



Gambar 4.18	Tampilan klasifikasi.....	57
Gambar 5.1	Format file hasil klasifikasi.....	59
Gambar 5.2	Citra korosi kelas crack.....	59
Gambar 5.3	Citra korosi kelas pit.....	59
Gambar 5.4	Citra awal.....	62
Gambar 5.5	Citra hasil pencerminan.....	62
Gambar 5.6	Kombinasi 1%.....	63
Gambar 5.7	Kombinasi 10%.....	63
Gambar 5.8	Kombinasi 20%.....	63
Gambar 5.9	Kombinasi 30%.....	63
Gambar 5.10	Kombinasi 40%.....	63
Gambar 5.11	Kombinasi 50%.....	63
Gambar 5.12	Kombinasi 60%.....	63
Gambar 5.13	Kombinasi 70%.....	63
Gambar 5.14	Kombinasi 80%.....	63
Gambar 5.15	Kombinasi 90%.....	63
Gambar 5.16	Kombinasi 100%.....	63
Gambar 5.17	Kurva nilai Euclidian.....	65
Gambar 5.17	Proses klasifikasi dan perhitungan waktunya.....	66

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Korosi merupakan sesuatu yang sangat penting dalam ilmu material dan muncul pada berbagai macam material dalam bentuk yang berbeda tergantung pada lingkungan sekitarnya. Sebagai hasil dari fenomena fisika dan kimia yang kompleks, korosi mempunyai sejumlah morfologi yang dapat dibedakan menjadi beberapa tipe.

Tugas akhir ini adalah untuk menunjukkan bahwa metode analisis tekstur sangat berguna untuk mengklasifikasikan tipe-tipe korosi tersebut, yang dalam hal ini dikonsentrasikan pada dua morfologi dasar : *pit formation* (lobang) dan *cracking* (pecahan). Karena dua macam morfologi tersebut dapat ditemukan pada material yang berbeda, lingkungan yang berbeda, dan terjadi dengan proses yang berbeda, maka citra yang menunjukkan morfologi sama dapat kelihatan sangat berbeda.

Penyebab utama terjadinya korosi adalah terlalu kompleks untuk digunakan dalam sistem pengenalan otomatis. Sehingga, hanya citra itu sendiri yang dapat mendefinisikan kelas yang ada.

Dengan adanya klasifikasi korosi tersebut akan diketahui tipe korosi yang terjadi pada suatu benda, sehingga dapat dilakukan langkah-langkah tertentu untuk mengantisipasinya sesuai dengan tipe korosi yang ada.



## 1.2 Perumusan Masalah

Unsur utama dalam pembuatan tugas akhir ini adalah ekstraksi *feature* dan klasifikasi.

### Ekstraksi *feature*

#### • Dekomposisi wavelet

Pada citra satu dimensi, transformasi wavelet dari suatu sinyal  $s(t)$  dibentuk dengan memproyeksikan  $s$  ke kumpulan wavelet yang membentuk basis ortogonal. Kumpulan ini terdiri dari dilatasi dan translasi *mother wavelet* tunggal. Hasil representasi berisi suatu sinyal terpisah untuk setiap skala resolusi.

Transformasi wavelet citra dua dimensi  $I(x,y)$  dapat dibentuk dengan menerapkan filter H dan L yang sama berurutan sepanjang baris dan kolom citra. Subcitra yang dihasilkan dari suatu operator dapat ditulis sebagai :

$$L^1(m,n) = [L_x * [L_y * I]](x,y)$$

$$D_1^1(m,n) = [L_x * [H_y * I]](x,y)$$

$$D_2^1(m,n) = [H_x * [L_y * I]](x,y)$$

$$D_3^1(m,n) = [H_x * [H_y * I]](x,y)$$

dengan  $*$  menyatakan operator konvolusi, sedangkan  $m, n$  merupakan koordinat sumbu x dan sumbu y.  $L^1$  merupakan citra yang dihasilkan oleh proses dekomposisi terhadap citra asli  $I$ . Citra detail  $D_1^1, D_2^1, D_3^1$  merupakan detail untuk arah vertikal, horizontal dan diagonal. Konvolusi yang pertama dilakukan terhadap kolom citra, konvolusi yang kedua dilakukan terhadap baris citra.



- Energi

Dekomposisi membedakan informasi dari skala yang berbeda. Energi konvensional dapat ditentukan dengan

$$E_j^i = \frac{1}{MN} \sum_{m=1}^M \sum_{n=1}^N (D_j^i(m, n))^2$$

Dengan  $M, N$  menyatakan ukuran dari subcitra.

Komponen vektor terdiri dari energi subcitra yang dihasilkan dari dekomposisi wavelet.

### Klasifikasi

Sebelum dilakukan proses klasifikasi, dilakukan terlebih dahulu proses pembelajaran. Pada tahap pembelajaran, contoh-contoh dari *training set* yang telah diketahui kelasnya digunakan untuk mengkompilasi pengetahuan tentang distribusi kelas. Hasil dari proses pembelajaran ini adalah bobot untuk setiap kelas, yang nantinya digunakan untuk melakukan proses klasifikasi.

Cara yang biasa dipakai untuk mengevaluasi performansi sistem adalah dengan membandingkan kelas yang sesungguhnya dengan kelas hasil klasifikasi terhadap citra masukan.

Pada tugas akhir ini, digunakan metode jaringan syaraf buatan, yaitu *Learning Vector Quantization (LVQ) network* sebagai dasar untuk melakukan klasifikasi.



### 1.3 Batasan Masalah

Tugas akhir ini dibatasi dengan beberapa kondisi sebagai berikut :

- Citra yang digunakan sebagai input adalah citra dengan level warna dan ukuran tertentu.
- Transformasi wavelet dilakukan sampai dengan level tertentu (dua level).
- Citra input mempunyai format file bitmap (file BMP).
- Citra input telah mengalami *preprocessing*.

### 1.4 Tujuan Tugas Akhir

- Membuat suatu perangkat lunak untuk mengklasifikasikan morfologi korosi suatu bahan.

### 1.5 Manfaat Tugas Akhir

- Mengklasifikasikan morfologi korosi suatu bahan yang dapat digunakan untuk menentukan langkah-langkah berikutnya.
- Sebagai bahan referensi untuk penelitian selanjutnya dalam bidang ilmu material, kecerdasan buatan dan citra digital.

### 1.6 Metodologi

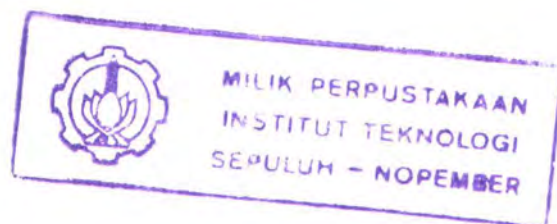
- Studi literatur
- Mempelajari transformasi wavelet multiresolusi, proses konvolusi, jaringan syaraf buatan (terutama *Learning Vector Quantization Network*).
- Desain program

- Perancangan secara teoretis terhadap perangkat lunak yang akan dibuat.
- Pembuatan program
- Implementasi desain program yang sudah dibuat sesuai dengan kebutuhan-kebutuhan yang ada.
- Tes dan evaluasi
- Menganalisis kinerja perangkat lunak yang telah dibuat, dengan melakukan tes terhadap input yang berbeda-beda. Sehingga dapat diketahui tingkat keakuratan perangkat lunak tersebut.

### 1.7 Sistematika Penulisan

Dalam penulisannya, tugas akhir ini dibagi dalam enam bab. Masing-masing bab akan menjelaskan masalah-masalah sebagai berikut :

- BAB I : Pendahuluan, menguraikan latar belakang, perumusan masalah, batasan masalah, tujuan, manfaat, metodologi yang digunakan, dan sistematika penulisan tugas akhir.
- BAB II : Dasar teori wavelet, yang menguraikan teori-teori dasar yang digunakan untuk menganalisis transformasi wavelet, seperti teori-teori tentang aljabar linear.
- BAB III : Klasifikasi pola, yang menjelaskan klasifikasi pola secara umum dan diikuti dengan penggunaan jaringan syaraf buatan, sesuai dengan metode yang dipakai dalam tugas akhir ini.





- BAB IV : Perancangan dan Implementasi, berisi penyusunan algoritma perangkat lunak, mulai dari input yang digunakan sampai output yang dihasilkan.
- BAB V : Uji coba dan analisis hasil, berisi analisis terhadap kinerja perangkat lunak sehingga diketahui tingkat keakuratannya.
- BAB VI : Kesimpulan dan saran, yang memberikan kesimpulan dan saran-saran untuk penelitian selanjutnya.

## BAB II

### DASAR TEORI WAVELET DAN MULTIRESOLUSI

#### 2.1 Wavelet

Wavelet merupakan fungsi matematika yang menampilkan data pada komponen frekuensi yang berbeda dan kemudian mengolah masing-masing komponen dengan resolusi yang sesuai dengan skalanya.

Wavelet telah menjadi sesuatu yang sangat berguna pada aplikasi-aplikasi ilmu pengetahuan dan teknologi, termasuk di dalamnya grafika komputer, pemampatan data dan pengolahan sinyal. Dalam perkembangannya, wavelet telah banyak digunakan dalam bidang-bidang yang lain.

Sebagai alat bantu, wavelet dapat menggambarkan sinyal asli secara matematis. Sinyal asli yang dimaksud di sini dapat berupa sebuah citra, kurva atau yang lain. Pada tugas akhir ini, transformasi wavelet dihubungkan dengan analisis multiresolusi. Mengingat teori dasar analisis multiresolusi adalah aljabar linear, berikut ini akan dijelaskan secara singkat konsep dasar aljabar linear yang berhubungan dengan analisis multiresolusi.

#### Ruang vektor

Jika  $V$  sebarang himpunan benda yang dua operasinya didefinisikan, yaitu penambahan dan perkalian dengan skalar (bilangan real). Penambahan tersebut dipahami untuk mengasosiasikan sebuah aturan dengan setiap pasang benda  $u$  dan  $v$



dalam  $V$ , yang mengandung elemen  $u + v$ , yang disebut jumlah  $u$  dan  $v$ . Dengan perkalian skalar diartikan aturan untuk mengasosiasikannya baik untuk setiap skalar  $k$  maupun setiap benda  $u$  pada  $V$  yang mengandung elemen  $ku$ , yang dinamakan perkalian skalar (*skalar multiple*)  $u$  oleh  $k$ . Jika aksioma-aksioma berikut dipenuhi oleh semua benda  $u, v, w$  pada  $V$  dan oleh semua skalar  $k$  dan  $l$ , maka dinamakan  $V$  sebuah ruang vektor (*vector space*) dan benda-benda pada  $V$  dinamakan vektor :

- a. jika  $u$  dan  $v$  adalah benda-benda pada  $V$ , maka  $u + v$  berada di  $V$
- b.  $u + v = v + u$
- c.  $u + (v + w) = (u + v) + w$
- d. ada sebuah benda  $0$  di  $V$  sehingga  $0 + u = u + 0 = u$  untuk semua  $u$  di  $V$
- e. untuk setiap  $u$  di  $V$ , ada sebuah benda  $-u$  di  $V$  yang dinamakan negatif  $u$  sehingga  $u + (-u) = (-u) + u = 0$
- f. jika  $k$  adalah sebarang skalar dan  $u$  adalah sebarang benda di  $V$ , maka  $ku$  berada di  $V$
- g.  $k(u + v) = ku + kv$
- h.  $(k + l)u = ku + lu$
- i.  $k(lu) = (kl)(u)$
- j.  $1u = u$

Sebuah ruang vektor dapat saja terkandung di ruang vektor yang lebih besar. Misalnya, garis dan bidang yang melalui titik asal adalah ruang vektor yang terkandung dalam ruang vektor yang lebih besar pada  $R^3$ .

Jika  $W$  adalah himpunan dari satu atau lebih vektor dari sebuah ruang vektor  $V$ , maka  $W$  adalah subruang dari  $V$  jika dan hanya jika kondisi-kondisi berikut berlaku :

- a. jika  $u$  dan  $v$  adalah vektor-vektor pada  $W$ , maka  $u + v$  terletak di  $W$
- b. jika  $k$  adalah sebarang skalar dan  $u$  adalah sebarang vektor pada  $W$ , maka  $ku$  berada di  $W$ .

### Basis

Jika  $V$  adalah sebarang ruang vektor dan  $S = \{v_1, v_2, \dots, v_r\}$  merupakan himpunan berhingga dari vektor-vektor pada  $V$ , maka  $S$  dinamakan basis untuk  $V$  jika

- a.  $S$  bebas linear;
- b.  $S$  merentang  $V$

Jika  $e_1 = (1, 0, 0, \dots, 0)$ ,  $e_2 = (0, 1, 0, \dots, 0)$ , ...,  $e_n = (0, 0, 0, \dots, 1)$ . Karena setiap vektor  $v = (v_1, v_2, \dots, v_n)$  pada  $R^n$  dapat dituliskan sebagai  $v = v_1 e_1 + v_2 e_2 + \dots + v_n e_n$ , maka  $S$  merentang  $R^n$  sehingga  $S$  adalah sebuah basis.

Vektor dalam sebuah basis untuk  $V$  disebut *span*  $V$ .

### Hasil Kali Dalam dan Orthogonalitas

Sebuah hasil kali dalam (*inner product*) pada ruang vektor real  $V$  adalah fungsi yang mengasosiasikan bilangan real  $\langle u, v \rangle$  dengan masing-masing pasangan vektor  $u$  dan  $v$  pada  $V$  sedemikian rupa sehingga aksioma-aksioma berikut dipenuhi untuk semua vektor  $u, v$ , dan  $w$  di  $V$  dan juga untuk semua skalar  $k$ .

- a.  $\langle u, v \rangle = \langle v, u \rangle$  (aksioma simetri)
- b.  $\langle u + v, w \rangle = \langle u, w \rangle + \langle v, w \rangle$  (aksioma penambahan)



c.  $\langle ku, v \rangle = k \langle u, v \rangle$  (aksioma kehomogenan)

d.  $\langle v, v \rangle \geq 0$ ; dan  $\langle v, v \rangle = 0$  (aksioma kepositifan)

jika dan hanya jika  $v = 0$

Salah satu sifat hasil kali dalam adalah orthogonalitas. Dua buah vektor  $u$  dan  $v$  dikatakan orthogonal, jika hasil kali dalamnya memenuhi  $\langle u, v \rangle = 0$ .

### Norma

Panjang sebuah vektor  $v$  sering dinamakan norma  $v$  dan dinyatakan dengan  $\|v\|$ . Jelaslah bahwa dari teorema Pythagoras bahwa norma vektor  $v = (v_1, v_2)$  di ruang dua adalah

$$\|v\| = \sqrt{v_1^2 + v_2^2}$$

Sedangkan norma vektor  $v$  di ruang 3 adalah

$$\|v\| = \sqrt{v_1^2 + v_2^2 + v_3^2}$$



### Ruang Hilbert

Ruang Hilbert merupakan ruang hasil kali dalam dari fungsi satu dimensi  $f(x)$ , yang dinyatakan sebagai  $L^2(R)$  [3]. Untuk  $f(x), g(x) \in L^2(R)$  hasil kali dalamnya didefinisikan sebagai :  $\langle g(u), f(u) \rangle = \int_{-\infty}^{+\infty} g(u)f(u)du$ .

Normalisasi untuk  $f(x)$  pada  $L^2(R)$  didefinisikan sebagai  $\|f\|^2 = \int_{-\infty}^{+\infty} |f(u)|^2 du$ .

Konvolusi dari dua fungsi,  $f(x), g(x) \in L^2(R)$  dinyatakan sebagai berikut :

Hal ini dikenal sebagai relasi skala-dua untuk fungsi penyekalaan dan deret  $p\{k\}$  yang disebut deret skala-dua dari  $\phi$ . Pembahasan dibatasi pada fungsi-fungsi penyekalaan yang hanya memiliki  $p_k$  yang tidak nol secara terbatas pada hubungan di atas.

Suatu fungsi  $\phi \in L^2\{R\}$  dikatakan menghasilkan analisis multiresolusi (*Multi Resolution Analysis* / MRA) jika fungsi tersebut menghasilkan sekumpulan deretan pada subruang tertutup  $V_0$  yang memenuhi sifat-sifat sebagai berikut :

1.  $\dots V_{-1} \subset V_0 \subset V_1 \subset V_2 \dots$
2.  $\overline{\bigcup_{j \in \mathbb{Z}} V_j} = L^2(R)$
3.  $\bigcap_{j \in \mathbb{Z}} V_j = \{0\}$
4.  $f(x) \in V_j \Leftrightarrow f(2x) \in V_{j+1} \quad j \in \mathbb{Z}$

## 2.2 Multiresolusi

### 2.2.1 Analisis Wavelet

Wavelet merupakan fungsi real  $t$ , dilambangkan dengan  $\Psi(t)$  dalam ruang fungsi  $L^2(R)$ . Fungsi ini dihasilkan oleh parameter dilatasi dan translasi yang dinyatakan dalam persamaan yang dikemukakan oleh Grosman dan Morlet sebagai berikut :

$$\psi_{a,b}(t) = |a|^{-\frac{1}{2}} \psi\left(\frac{t-b}{a}\right)$$

dengan  $a$  adalah parameter dilatasi dan  $b$  adalah parameter translasi. Fungsi  $\Psi$  disebut *mother wavelet* dan harus memenuhi kondisi [5] :

$$\int_{-\infty}^{\infty} \psi(t) dt = 0$$



Transformasi wavelet dibedakan menjadi dua tipe, yaitu transformasi wavelet kontinu dan transformasi wavelet diskret. Faktor yang membedakan dua tipe transformasi tersebut terletak pada nilai parameter translasi dan parameter dilatasi. Secara umum bentuk transformasi wavelet dinyatakan sebagai berikut [5] :

$$(T_{\text{wav}} f)(a, b) = \langle f, \psi_{a,b}(x) \rangle$$

$$(T_{\text{wav}} f)(a, b) = |a|^{-1/2} \int_{-\infty}^{\infty} f(x) \psi\left(\frac{x-b}{a}\right) dx$$

Persamaan ini menunjukkan bahwa transformasi wavelet  $f(x) \in L^2(R)$  satu dimensi merupakan bentuk integral konvolusi antara fungsi  $f(x)$  dengan fungsi wavelet  $\psi_{a,b}(x)$ .

#### 2.2.1.1 Transformasi Wavelet Kontinu

Transformasi wavelet kontinu ditentukan oleh nilai parameter dilatasi  $a$  dan parameter translasi  $b$  yang bervariasi secara kontinu, dimana  $a, b \in R$  dengan  $a \neq 0$ .

#### 2.2.1.2 Transformasi Wavelet Diskret

Transformasi wavelet diskret dapat diperoleh dari transformasi wavelet kontinu dengan mengambil nilai parameter  $a$  dan  $b$  yang diskret saja. Transformasi ini dibedakan menjadi dua jenis, yaitu :

- Sistem diskret redundan

Diskretisasi parameter-parameter tersebut adalah  $a = a_0^{-m}$  dan  $b = nb_0 a_0^{-m}$ ; dimana  $m, n \in Z$  dan ditentukan suatu nilai tertentu untuk  $a_0 > 1$  dan  $b_0 > 0$  (tertentu) akibatnya fungsi wavelet jenis ini tergantung pada  $a_0, b_0$ , fungsi waveletnya terbentuk :

$$\psi_{a,b}(x) = a_0^{-m/2} \psi\left(x - nb_0 a_0^{-m}\right)$$

$$\psi_{a,b}(x) = a_0^{-m/2} \psi\left(a_0^m x - nb_0\right)$$

Transformasi ini dapat menyebabkan adanya redudansi pada penggambaran fungsi aslinya sebab dimungkinkan untuk menghitung nilai transformasi waveletnya secara pendekatan tanpa perlu perhitungan secara tepat.

- Transformasi wavelet basis orthonormal

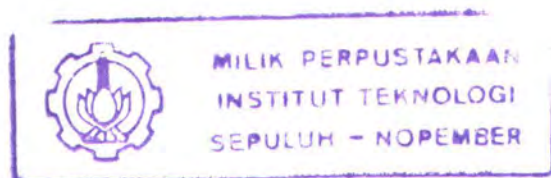
Dengan memiliki nilai-nilai yang sangat khusus untuk mother wavelet  $\psi$  dan  $a, b$ ; maka dapat diperoleh suatu nilai-nilai  $\psi_{m,n}$  yang memiliki basis orthonormal pada  $L^2(R)$ . Umumnya dipilih parameter  $a_0 = 2$  dan  $b_0 = 1$  [5]. Bentuk fungsi waveletnya dapat dinyatakan sebagai berikut :

$$\psi_{m,n}(x) = 2^{m/2} \psi(2^m x - n)$$

dimana  $m, n \in \mathbb{Z}$ .

Salah satu contoh fungsi  $\psi$  yang menyebabkan  $\psi_{m,n}$  pada persamaan di atas memiliki basis orthonormal pada  $L^2(R)$  adalah fungsi Haar, yaitu :

$$\psi(x) = \begin{cases} +1 & \text{jika } 0 \leq x \leq \frac{1}{2} \\ -1 & \text{jika } \frac{1}{2} < x \leq 1 \\ 0 & \text{untuk lainnya} \end{cases}$$



### 2.2.2 Analisis Multiresolusi

Transformasi multiresolusi telah dipelajari untuk aplikasi ilmu komputer sejak Rosenfeld dan Thurston [2] melakukan deteksi tepi multiskala. Pada resolusi yang berbeda, detail suatu citra mempunyai karakteristik yang berbeda pula. Untuk



mengolah perbedaan tersebut, dilakukan pengambilan informasi yang berbeda dari suatu citra pada dua resolusi yang berbeda. Jika terdapat urutan resolusi yang semakin bertambah  $(r_j)_{j \in \mathbb{Z}}$ , maka detail dari  $f(x)$  pada resolusi  $r_j$  didefinisikan sebagai perbedaan informasi antara aproksimasi  $f(x)$  pada resolusi  $r_{j-1}$  dan pendekatan pada resolusi  $r_j$ .

Jika  $A_{2^j}$  adalah operator dimana pendekatan sinyal pada resolusi  $2^j$  dan terdapat fungsi  $f(x)$  yang mempunyai energi :  $f(x) \in L^2(R)$ , maka dapat ditetapkan bahwa :

- $A_{2^j}$  merupakan operator linear. Jika  $A_{2^j} f(x)$  merupakan pendekatan terhadap fungsi  $f(x)$  pada resolusi  $2^j$ , maka  $A_{2^j} f(x)$  tidak berubah jika melakukan pendekatan lagi pada resolusi  $2^j$ . Operator  $A_{2^j}$  merupakan operator proyeksi pada ruang vektor  $V_{2^j} \subset L^2(R)$ . Ruang vektor  $V_{2^j}$  dapat diartikan sebagai himpunan dari semua pendekatan yang mungkin pada resolusi  $2^j$  dari fungsi yang ada pada  $L^2(R)$ .
- Dari semua fungsi pendekatan pada resolusi  $2^j$ ,  $A_{2^j} f(x)$  merupakan fungsi yang paling dekat ke  $f(x)$ .

$$\forall g(x) \in V_{2^j}, \|g(x) - f(x)\| \geq \|A_{2^j} f(x) - f(x)\|$$

Sehingga operator  $A_{2^j}$  merupakan proyeksi ortogonal pada ruang vektor  $A_{2^j}$ .

- Pendekatan sinyal pada resolusi  $2^{j+1}$  berisi semua informasi penting untuk melakukan perhitungan pada sinyal yang sama dengan resolusi yang lebih kecil,  $2^j$ . Karena  $A_{2^j}$  merupakan operator proyeksi pada  $V_{2^j}$  maka prinsip ini ekuivalen dengan persamaan berikut :

$$\forall j \in \mathbb{Z}, \quad V_{2^j} \subset V_{2^{j+1}}$$

Suatu operasi pendekatan adalah sama pada semua resolusi. Ruang fungsi pendekatan diturunkan dari satu sama lain dengan menyekala setiap fungsi pendekatan menggunakan rasio dari nilai resolusinya.

$$\forall j \in \mathbb{Z}, \quad f(x) \in V_{2^j} \Leftrightarrow f(2x) \in V_{2^{j+1}}$$

- Pendekatan  $A_{2^j} f(x)$  terhadap sinyal  $f(x)$  dapat dikarakterisasikan oleh  $2^j$  sampel per unit panjang. Jika  $f(x)$  ditranslasikan dengan panjang sebanding dengan  $2^j$ ,  $A_{2^j} f(x)$  ditranslasikan oleh jumlah yang sama dan dikarakterisasi oleh sampel yang sama. Terdapat prinsip-prinsip untuk  $j = 0$ , yaitu translasi matematika terdiri dari :

- karakterisasi diskret

terdapat isomorfisme  $I$  dari  $V_1$  ke  $l^2(\mathbb{Z})$

- translasi pendekatan

$$\begin{aligned} \forall k \in \mathbb{Z}, A_1 f_k(x) \\ = A_1 f(x-k), \quad f_k(x) = f(x-k) \end{aligned}$$

- translasi sampel

$$I(A_1 f(x)) = (\alpha_i)_{i \in \mathbb{Z}} \Leftrightarrow I(A_1 f_k(x)) = (\alpha_{i-k})_{i \in \mathbb{Z}}$$

- ketika melakukan perhitungan suatu pendekatan  $f(x)$  pada resolusi  $2^j$ , beberapa informasi mengenai  $f(x)$  hilang. Dengan adanya peningkatan resolusi menjadi  $+\infty$  pendekatan sinyal harus konvergen terhadap sinyal asli. Kebalikannya, dengan adanya resolusi berkurang menjadi nol, pendekatan sinyal berisi sedikit informasi dan konvergen ke nol.



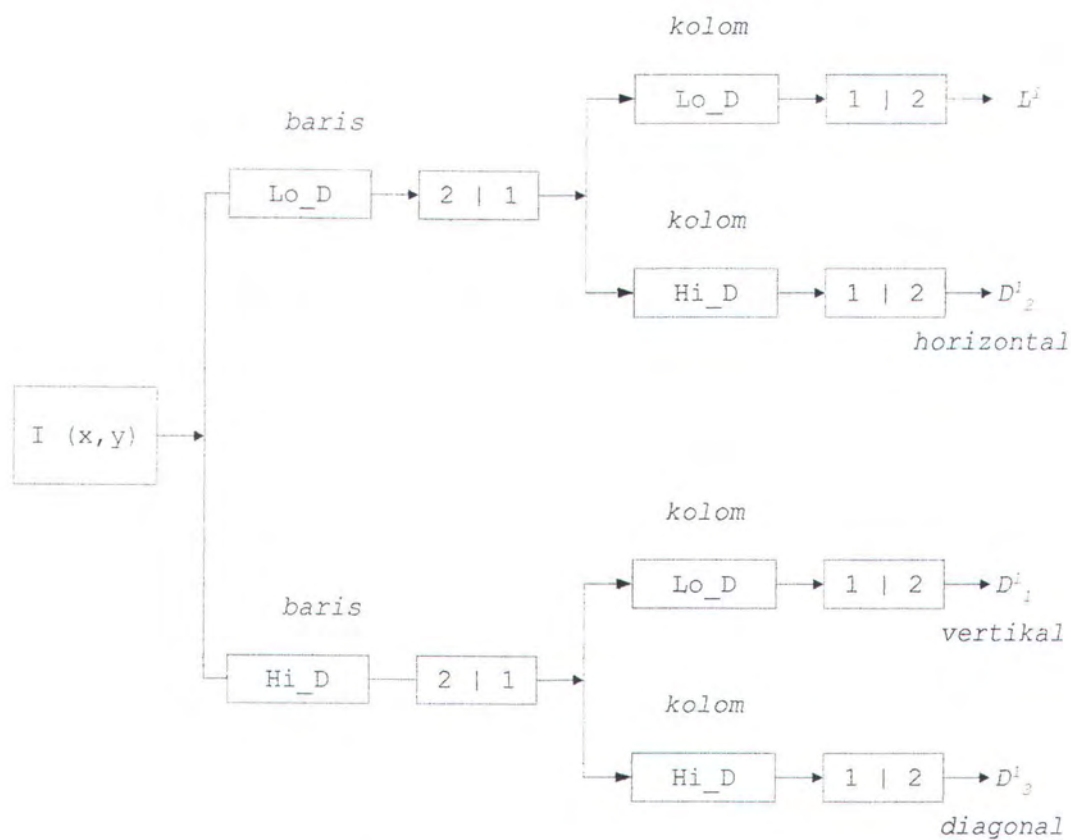
### 2.3 Analisis Citra berdasarkan Wavelet

Untuk mengaplikasikan transformasi wavelet pada citra, digunakan pengembangan metode untuk dua dimensi. Transformasi wavelet pada citra dua dimensi  $I(x,y)$  dapat dibentuk dengan mengaplikasikan filter H dan L berurutan sepanjang kolom dan baris citra. Subcitra yang dihasilkan adalah :

$$\begin{aligned}L^1(m,n) &= [L_x * [L_y * I]](x,y) \\ D_1^1(m,n) &= [L_x * [H_y * I]](x,y) \\ D_2^1(m,n) &= [H_x * [L_y * I]](x,y) \\ D_3^1(m,n) &= [H_x * [H_y * I]](x,y)\end{aligned}$$

dengan  $*$  menyatakan operator konvolusi, sedangkan  $m,n$  merupakan koordinat sumbu x dan sumbu y.  $L^1$  merupakan citra yang dihasilkan oleh proses dekomposisi terhadap citra asli  $I$ . Citra detail  $D_1^1, D_2^1, D_3^1$  merupakan detail untuk arah vertikal, horizontal dan diagonal. Konvolusi yang pertama dilakukan terhadap kolom citra, konvolusi yang kedua dilakukan terhadap baris citra. Setelah itu subsampling pada kedua arah dapat dilakukan. Proses dekomposisi dilakukan seperti pada gambar 2.1

Sebagaimana pada kasus satu dimensi, dengan iterasi menggunakan lowpass pada subcitra  $L^{i-1}$  ( $L^i, D_1^i, D_2^i, D_3^i$ ) pada level yang berbeda dapat dilakukan. Hasilnya adalah suatu dekomposisi wavelet dengan detail citra untuk skala dan orientasi yang berbeda. Jika hanya pada  $L^i$  dilakukan dekomposisi, maka didapatkan wavelet standar (StW)/piramid. Dan jika pada semua subcitra dilakukan dekomposisi didapatkan dekomposisi wavelet paket (WP).



Gambar 2.1 Proses dekomposisi

2 | 1

*downsample* columns : hapus kolom yang berindeks genap

1 | 2

*downsample* baris : hapus baris yang berindeks genap

baris

x

konvolusi dengan filter x pada baris citra

kolom

x

konvolusi dengan filter x pada kolom citra



## 2.4 Analisis Tekstur berdasarkan Wavelet

### *Feature*

Feature merupakan bagian data yang dapat diolah, misalnya intensitas sinyal. Feature bisa bersifat simbolik seperti warna; numerik seperti halnya berat. Features dapat dihasilkan dari penerapan *feature extraction algorithm* terhadap data yang dimasukkan. Feature bisa mempunyai level entitas yang lebih tinggi, misalnya : *geometric descriptor* dari suatu daerah citra atau penampakan suatu citra sebagai obyek tiga dimensi.

Dalam pengolahan feature perlu diperhatikan bahwa :

- Proses pengolahan mungkin diperlukan untuk ekstraksi feature
- Ekstraksi feature mungkin berisi kesalahan atau *noise*

Feature bisa dinyatakan sebagai variabel kontinu, diskret, atau biner-diskret. Feature biner digunakan untuk menyatakan ada atau tidaknya feature tertentu.

Proses ekstraksi dan seleksi feature harus memenuhi beberapa hal, yaitu :

- Dapat dilakukan perhitungan
- Dapat menghasilkan sistem pengenalan pola yang “baik”
- Mengurangi persoalan tanpa menghilangkan informasi yang penting

### **Seleksi Feature**

Seleksi feature merupakan proses memilih masukan yang digunakan untuk sistem klasifikasi pola. Batasan-batasan perhitungan akan menentukan feature yang dipakai. Seleksi feature bisa dilakukan dengan menggunakan perhitungan matematika atau dengan cara simulasi.

## Ekstraksi feature

Jika terdapat data masukan dua dimensi  $f(x, y)$ , maka himpunan feature

didefinisikan sebagai :  $m_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x, y) dx dy \quad p, q = 0, 1, 2, \dots$

dengan x merupakan koordinat sumbu X, dan y merupakan koordinat sumbu Y.

*Central moment* didefinisikan sebagai :

$$\mu_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left( x - \hat{x} \right)^p \left( y - \hat{y} \right)^q f(x, y) dx dy$$
$$\hat{x} = m_{10} / m_{00}, \hat{y} = m_{01} / m_{00}$$

Algoritma yang paling sesuai untuk seleksi feature adalah *sequential forward floating selection (FFFS)*[6]. Proses inisialisasi dilakukan dengan mengambil satu feature yang terbaik dari himpunan. Kemudian, menambahkan feature terbaik dari sisanya, atau menghapus yang terjelek dari himpunan, sampai dicapai keadaan yang paling optimal. Keadaan terbaik dan terjelek ditentukan dengan mengukur perbedaan antara performansi klasifikasi dengan atau tanpa feature. Kondisi terbaik diperoleh jika terdapat pengaruh yang besar dengan adanya sisipan dan terjelek akan diperoleh jika feature tersebut tidak banyak mempengaruhi hasil yang didapatkan..

Pengambilan feature tidak harus sebanyak mungkin. Meskipun dengan adanya banyak feature mungkin membawa banyak nilai, akan menimbulkan tingkat kerumitan yang lebih besar. Untuk itu harus ditentukan penggunaan feature yang paling efektif.

Dari berbagai kemungkinan cara ekstraksi feature yang ada, kuadrat nilai paling sering digunakan [6]. Yaitu dengan persamaan :



$$E_j^i = \frac{1}{MN} \sum_{m=1}^M \sum_{n=1}^N (D_j^i(m, n))^2$$

dengan  $M, N$  menyatakan ukuran subcitra.

### Rotation Invariance

Berdasar pada fakta bahwa transformasi dilakukan sepanjang arah vertikal dan horisontal, rotasi kecil dapat mempunyai efek yang serius terhadap hasil.

Terdapat metode dimana feature wavelet ditransformasikan menjadi feature *rotation invariant*. Pada skala  $i$ , energi yang diasosiasikan dengan detail subcitra  $D_j^i$  dapat diartikan sebagai energi untuk satu arah. Pada subcitra didapatkan bahwa persamaan :

$$E_j^i(m, n) = (D_j^i(m, n))^2$$

yang dinotasikan dengan  $e_j^i$  merepresentasikan energi lokal untuk satu arah. Dengan menjumlahkan tiga energi lokal  $e_{j=1,2,3}^i$  akan didapatkan total energi per piksel (untuk skala ke- $i$ ).

Didapatkan bahwa :

$$E_{tot}^i(m, n) = e_1^i + e_2^i + e_3^i$$

dengan substitusi didapatkan :

$$E_{tot}^i(m, n) = (D_1^i(m, n))^2 + (D_2^i(m, n))^2 + (D_3^i(m, n))^2$$

dengan menjumlahkan global feature global subcitra per skala :

$$E_{tot}^i = \frac{1}{MN} \sum_{m=1}^M \sum_{n=1}^N E_{tot}^i(m, n)$$

dari  $e_{j=1,2,3}^i$  dapat dilakukan perhitungan untuk mendapatkan *orientational anisotropy* atau arah energi, yaitu :

$$Orian^i(x, y) = \frac{1}{E_{tot}^i(x, y)} \sqrt{(e_1^i - e_2^i)^2 + (e_1^i - e_3^i)^2 + (e_2^i - e_3^i)^2}$$

$Orian^i(x, y)$  merupakan piksel anisotropi dari energi yang memberikan pengukuran yang efektif terhadap distribusi energi berkaitan dengan arah. Dengan menjumlahkan feature global subcitra didapatkan :

$$Orian^i = \frac{1}{MN} \sum_{m=1}^M \sum_{n=1}^N Orian(m, n)$$

Untuk dekomposisi wavelet dengan level  $d$  didapatkan  $2d + 1$  dimensi, yang berisi  $E_{tot}^i$  dan  $Orian^i$  untuk setiap skala  $i$ , ditambah komponen tambahan untuk energi dari low pass  $L^d$ .

Untuk dekomposisi wavelet paket didapatkan feature sebanyak  $2^d$  untuk orientasi paket wavelet (OWP). Jika konsep orian diterapkan didapatkan feature sebanyak  $2^{d+1}$ , dengan menghitung Orian untuk setiap tiga subcitra yang berasal dari subcitra yang mempunyai level yang sama sebelumnya.

Ekivalensi dari subcitra dengan  $d = 2$  adalah seperti pada gambar 2.2 dan 2.3. Subcitra yang mempunyai tanda yang sama menyatakan ekivalensi subcitra tersebut.





$L^2(D^1_1)$	$D^2_1$	$D^1_1$
$D^2_2$	$D^2_3$	
$D^1_2$		$D^1_3$

Gambar 2.2 Wavelet standar

$L^2$	$D^2_1$	$L^2(D^1_1)$	$D^2_1(D^1_1)$
$D^2_2$	$D^2_3$	$D^2_2(D^1_1)$	$D^2_3(D^1_1)$
$L^2(D^1_2)$	$D^2_1(D^1_3)$	$L^2(D^1_3)$	$D^2_1(D^1_3)$
$D^2_2(D^1_3)$	$D^2_3(D^1_3)$	$D^2_2(D^1_3)$	$D^2_3(D^1_3)$

Gambar 2.3 Wavelet paket





### **BAB III**

## **KLASIFIKASI POLA**



## BAB III

### KLASIFIKASI POLA

#### 3.1 Definisi

*Klasifikasi* merupakan pengelompokan data masukan ke dalam satu atau lebih kelas yang telah ditentukan berdasarkan ekstraksi feature yang diperlukan atau atribut dan pengolahannya atau analisis terhadap atribut-atributnya.

*Pengenalan* adalah kemampuan untuk melakukan klasifikasi. Pengenalan pola sering kali dirumuskan dengan menambahkan satu kelas tambahan, yaitu kelas *unclassifiable*.

*Kelas pola* adalah himpunan pola. Hal yang penting pada beberapa aplikasi klasifikasi pola adalah indentifikasi atribut yang diperlukan (misal : feature) dan menentukan perhitungan yang baik terhadap proses pengenalan/klasifikasi.

*Preprocessing* adalah proses pemfilteran atau transformasi data mentah menjadi data yang lebih “baik” dan memperkecil adanya noise.

*Noise* adalah keadaan yang tidak ideal, termasuk di dalamnya adalah :

- kesalahan dalam pemrosesan awal;
- kesalahan dalam ekstraksi feature;
- kesalahan dalam pembelajaran data.

#### **Pola (*Pattern*)**

Kebanyakan informasi yang berada di sekeliling kita mempunyai bentuk yang berpola. Informasi tersebut seringkali diklasifikasikan dan dideskripsikan salah,

padahal kemampuan ini bisa diotomatisasi. Informasi-informasi yang mempunyai pola sama dapat dimasukkan ke dalam kelas/kelompok yang sama, sehingga dari informasi yang jumlahnya sangat banyak tersebut dapat dikenali/diklasifikasikan dengan mudah.

Pengenalan pola pada dasarnya berdasar pada *pola (pattern)*. Suatu pola dapat berupa himpunan hasil pengukuran atau penelitian yang dinyatakan dalam vektor atau matriks. Pola bisa diubah dari satu bentuk data ke bentuk yang lain, misalnya : dari citra dua dimensi menjadi vektor.

### 3.2 Pengklasifikasi / Classifier

Pengklasifikasi akan memecah ruang feature menjadi kelas yang disebut daerah keputusan (*decision region*). Daerah keputusan tersebut harus bersifat *nonoverlapping*. Batas masing-masing daerah disebut dengan batas keputusan (*decision boundary*).

Dengan sudut pandang tersebut, klasifikasi vektor  $\underline{x}$  menjadi cukup sederhana, yaitu menentukan daerah keputusan (dalam  $R^d$ ) dimana  $\underline{x}$  berada dan memasukkan  $\underline{x}$  ke dalam kelas tersebut. Yang menjadi permasalahan dalam hal ini adalah menentukan daerah keputusan dari  $\underline{x}$ .

Terdapat pengklasifikasi berdasarkan fungsi diskriminan (*discriminant function*). Pada kasus kelas  $c$  fungsi diskriminan dinyatakan dengan  $g_i(x)$ ,  $i = 1, 2, \dots, c$ , digunakan untuk membagi  $R^d$  sebagai berikut :

- Masukkan  $\underline{x}$  ke dalam kelas  $w_m$  (daerah  $R_m$ ) dimana  $g_m(\underline{x}) > g_i(\underline{x}) \forall i = 1, 2, \dots, c$  dan  $l \neq m$ .

Pada kasus dimana  $g_k(\underline{x}) = g_l(\underline{x})$  akan terbentuk batas keputusan.



Bentuk khusus dari fungsi diskriminan yang penting adalah fungsi diskriminan linear (*linear discriminant function*) :  $g_i(\underline{x}) = \underline{w}_i^T \underline{x} + w_{0i}$  dimana  $\underline{w}_i$  adalah vektor bobot  $d \times 1$  yang digunakan untuk kelas ke- $i$ . Permasalahan yang muncul adalah bagaimana menggunakan statistik, sintatik, atau jaringan syarat tiruan untuk mengimplementasikan fungsi diskriminan.

### **Supervised classifier**

Untuk mengaplikasikan pengklasifikasi ke dalam persoalan yang ada, pengetahuan mengenai distribusi kelas harus ditentukan dari himpunan pelatihan (*training set*) dari tahap pembelajaran (*learning stage*). Pengetahuan ini kemudian ditransformasikan ke dalam aturan klasifikasi. Contoh data yang belum pernah dipakai, dimasukkan ke dalam pengklasifikasi dan dilakukan pengecekan terhadap hasil yang ada (tahap percobaan / *test stage*).

Pemisahan kelas pada ruang feature menentukan keberhasilan pengklasifikasi. Ruang feature adalah multidimensi. Jika fungsi densitas probabilitas kelas diketahui, pengklasifikasi Bayesian akan memaksimalkan kebenaran klasifikasi, karena daerah ruang masukan akan diasosiasikan dengan kelas yang mempunyai probabilitas yang besar[9].

Pada praktiknya, fungsi probabilitas tidak diketahui. Seberapa baik tingkat estimasinya tergantung pada bagaimana himpunan pelatihan (*training set*) direpresentasikan untuk semua data. Hal ini terutama ditentukan oleh ukuran himpunan pelatihan dan dimensi ruang feature.

tertentu. Komponen otak manusia yang “digunakan” pada jaringan syaraf tiruan tersebut adalah :

- *Neuron*
- *Synapsis*
- Sel pendukung

Neuron merupakan elemen utama sistem syaraf. Sedangkan sistem syaraf tersusun dari sekitar lima juta neuron. Neuron mempunyai fungsi-fungsi khusus, yaitu : menerima sinyal yang datang dari neuron tetangga, mengintegrasikan sinyal-sinyal tersebut, menimbulkan pulsa syaraf, menghubungkan pulsa tersebut dan mentransmisikannya ke neuron lain yang dapat menerima.

Synapsis diperlukan untuk berkomunikasi antara sel-sel syaraf. Sinyal dipancarkan sepanjang sel secara elektrik tetapi tidak terdapat hubungan langsung antara dua sel syaraf. Mereka terpisah oleh ruang yang disebut *synaptic gap*, yang tidak dapat dilewati pulsa listrik.

Sel pendukung, sebagai bagian terpisah dari neuron, mempunyai banyak fungsi, seperti untuk penyaluran nutrisi.

### **Fase desain**

Untuk mendesain pengklasifikasi pola berbasiskan jaringan syaraf, yang terutama harus diperhatikan adalah :

- Jaringan tersebut dapat dilatih untuk membentuk klasifikasi yang diinginkan, dan tidak terjadi ambiguitas.
- Struktur jaringan yang cocok sehingga diasumsikan bahwa semua masalah dapat dipecahkan.



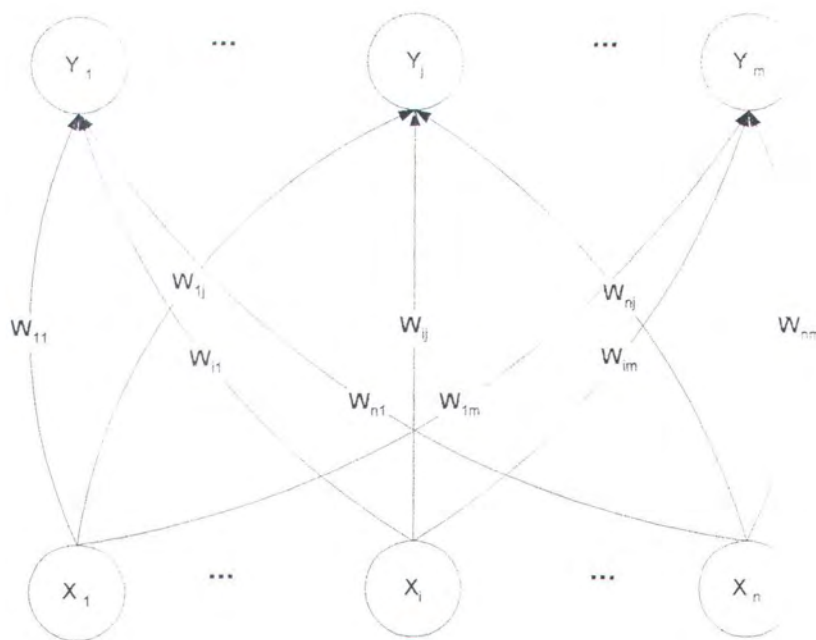
### 3.3 Learning Vector Quantization (LVQ)

LVQ merupakan metode untuk mengklasifikasikan pola, di mana masing-masing output merepresentasikan kelas atau kategori tertentu. Vektor bobot dari unit output sering disebut sebagai vektor *reference (codebook)* untuk kelas dari unit yang direpresentasikan. Selama training, akan terjadi perubahan vektor bobot.

Setelah training, LVQ akan mengklasifikasikan vektor input ke dalam kelas yang sama dengan unit output yang mempunyai vektor bobot terdekat dengan vektor input.

#### 3.3.1 Arsitektur

Arsitektur LVQ adalah sebagai berikut :



Gambar 3.2 Arsitektur LVQ





## **BAB IV**

# **PERANCANGAN DAN PEMBUATAN PERANGKAT LUNAK**



## BAB IV

### PERANCANGAN DAN PEMBUATAN

#### PERANGKAT LUNAK

##### 4.1 Pendahuluan

Pada bab ini akan dibahas perancangan dan pembuatan perangkat lunak untuk mengklasifikasikan citra korosi ke dalam morfologi tertentu, yaitu *pit* dan *crack*. Perangkat lunak dibuat pada sistem operasi Windows 98 dengan menggunakan kompiler Borland Delphi 4.0.

Klasifikasi citra korosi ini terdiri dari dua bagian utama, yaitu dekomposisi citra dan proses pengolahan menggunakan jaringan syaraf, yaitu *LVQ (Learning Vector Quantization)*. Proses pengolahan ini terdiri atas inisialisasi, pelatihan, dan klasifikasi.

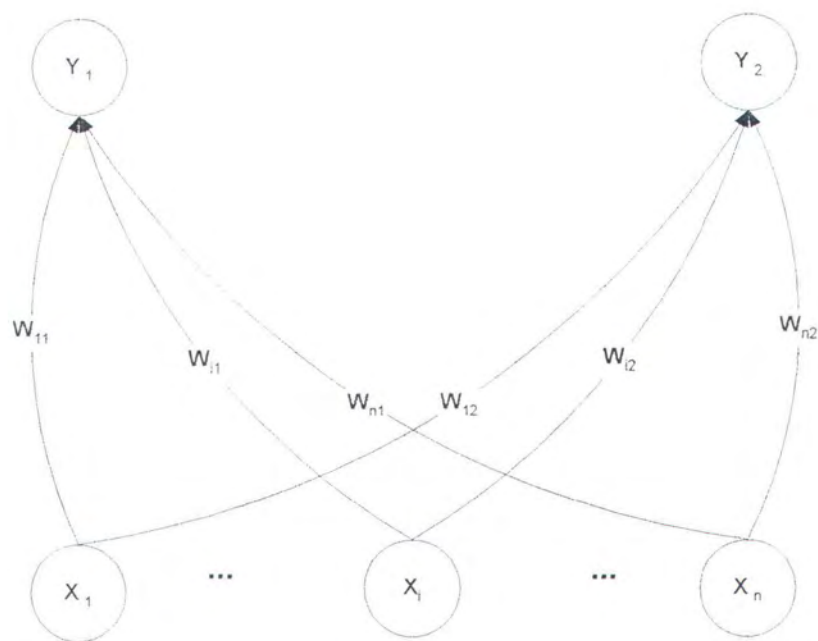
##### 4.2 Perancangan Perangkat Lunak dan Tujuannya

Tujuan perancangan perangkat lunak ini adalah untuk mempermudah implementasi algoritma yang dipakai. Perancangan ini terdiri dari perancangan arsitektur jaringan syaraf, perancangan data, dan perancangan proses.

###### 4.2.1 Perancangan Arsitektur Jaringan Syaraf

Arsitektur *LVQ* pada dasarnya sama dengan arsitektur *Kohonen Self-Organizing Map*. Yang membedakannya, bahwa untuk *LVQ* masing-masing keluaran mempunyai kelas yang telah didefinisikan. Dalam hal ini, terdapat dua buah keluaran (*output*), yaitu kelas *pit* dan kelas *crack*. Sedangkan banyaknya masukan

(input) tergantung dari jenis dekomposisi dan pemilihan feature yang dipakai. Dalam bentuk diagram, arsitektur *LVQ* ini dapat dinyatakan dengan gambar 4.1.



Gambar 4.1 Arsitektur jaringan syaraf

- $x$  vektor masukan ( $x_1, \dots, x_i, \dots, x_n$ )
- $y$  vektor keluaran ( $y_1, y_2$ )
- $w_{ij}$  vektor bobot untuk unit keluaran ke- $j$  ( $w_{1j}, \dots, w_{ij}, \dots, w_{nj}$ )

### 4.2.2 Perancangan Data

Data yang digunakan untuk implementasi perangkat lunak ini dibagi menjadi tiga bagian utama, yaitu data masukan, data proses dan data keluaran.



#### 4.2.2.1 Data Masukan

Perangkat lunak ini menggunakan data masukan berupa file citra bertipe bitmap dengan ukuran 128 x 128 piksel. Selanjutnya, nilai setiap piksel akan dipindahkan pada variabel bertipe TList.

#### 4.2.2.2 Data Proses

Selama proses berlangsung, digunakan beberapa macam data. Yaitu, data proses inisialisasi, data proses pelatihan dan data proses klasifikasi.

- Proses inisialisasi, pelatihan dan klasifikasi dengan masukan citra, akan melalui proses dekomposisi. Data proses dekomposisi disimpan dalam field bertipe array dua dimensi bertipe TList yang terdapat pada kelas TFeature.
- Data hasil proses inisialisasi berupa data bobot disimpan dalam bentuk vektor. Data ini disimpan dalam file sebagai file teks, yang selanjutnya digunakan sebagai data awal untuk pelatihan atau dapat digunakan untuk klasifikasi secara langsung.
- Data masukan untuk proses pelatihan atau klasifikasi disimpan dalam bentuk array dua dimensi bertipe single.
- Data hasil proses pelatihan berupa array dua dimensi bertipe single sebagai nilai-nilai bobot untuk proses pelatihan selanjutnya atau untuk proses klasifikasi.

#### 4.2.2.3 Data Keluaran

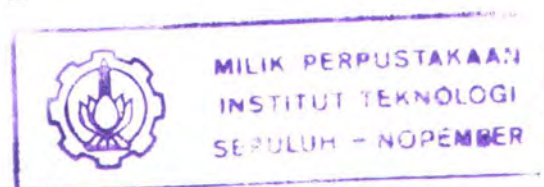
Data keluaran yang dihasilkan dari proses klasifikasi berupa vektor yang selanjutnya dicetak pada file teks. Informasi yang disimpan adalah kelas untuk setiap data masukan.

#### 4.2.3 Perancangan Proses

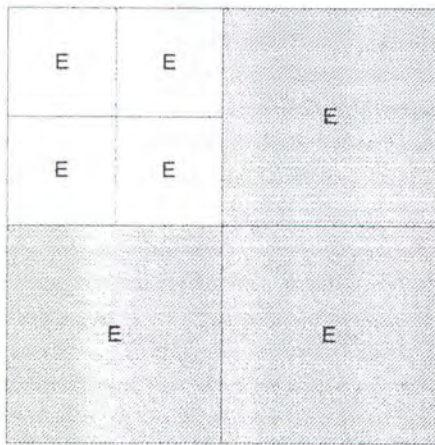
Proses pada sistem ini terdiri atas proses dekomposisi terhadap file citra sebagai masukan dan proses pengolahan jaringan syaraf, yaitu proses inisialisasi, proses pelatihan dan proses klasifikasi. Proses dekomposisi merupakan proses awal untuk pengolahan jaringan syaraf. Masing-masing feature tersebut digunakan sebagai masukan untuk pengolahan jaringan syaraf.

Seperti yang telah dijelaskan pada bab II, bahwa dekomposisi dapat dilakukan dengan menggunakan dua metode, yaitu : wavelet standar dan wavelet paket. Dari dua metode tersebut, diambil feature dengan dua metode pula yaitu : feature energi serta feature energi dan orian. Sehingga didapatkan empat metode pengambilan feature yang berbeda, yaitu (untuk dekomposisi level dua) :

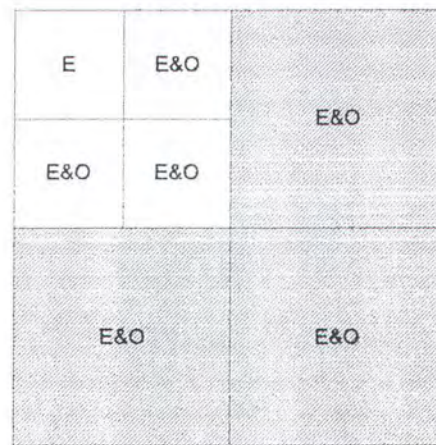
- OW, dekomposisi dengan menggunakan wavelet standar dengan feature energi saja, didapatkan 3 feature (gambar 4.2)
- Orian W, dekomposisi dengan menggunakan wavelet standar dengan feature energi dan orian, didapatkan 5 feature (gambar 4.3)
- WP, dekomposisi dengan menggunakan wavelet paket dengan feature energi saja, didapatkan 4 feature (gambar 4.4)
- Orian WP, dekomposisi dengan menggunakan wavelet paket dengan feature energi dan orian, didapatkan 8 feature (gambar 4.5).



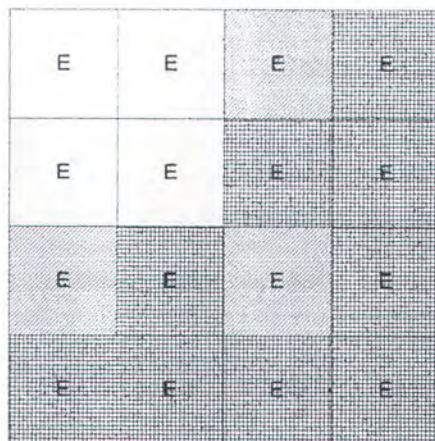




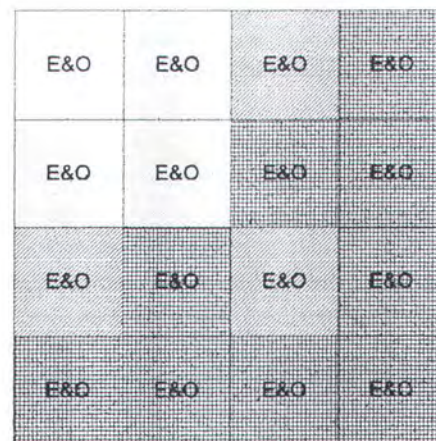
Gambar 4.2 Dekomposisi 3 feature



Gambar 4.3 Dekomposisi 5 feature



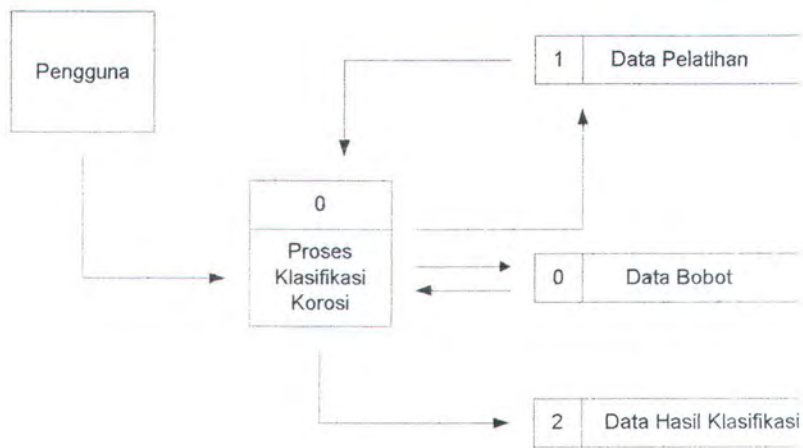
Gambar 4.4 Dekomposisi 4 feature



Gambar 4.5 Dekomposisi 8 feature

Notasi 'E' menyatakan bahwa pada subcitra yang ekuivalen dilakukan penghitungan energi. Notasi 'E&O' menyatakan bahwa pada subcitra yang ekuivalen dilakukan penghitungan energi dan orian. Ekuivalensi subcitra dinyatakan dengan penggambaran subcitra yang sama.

Untuk memperjelas alur proses perangkat lunak ini digunakan diagram, seperti DFD pada gambar 4.6.



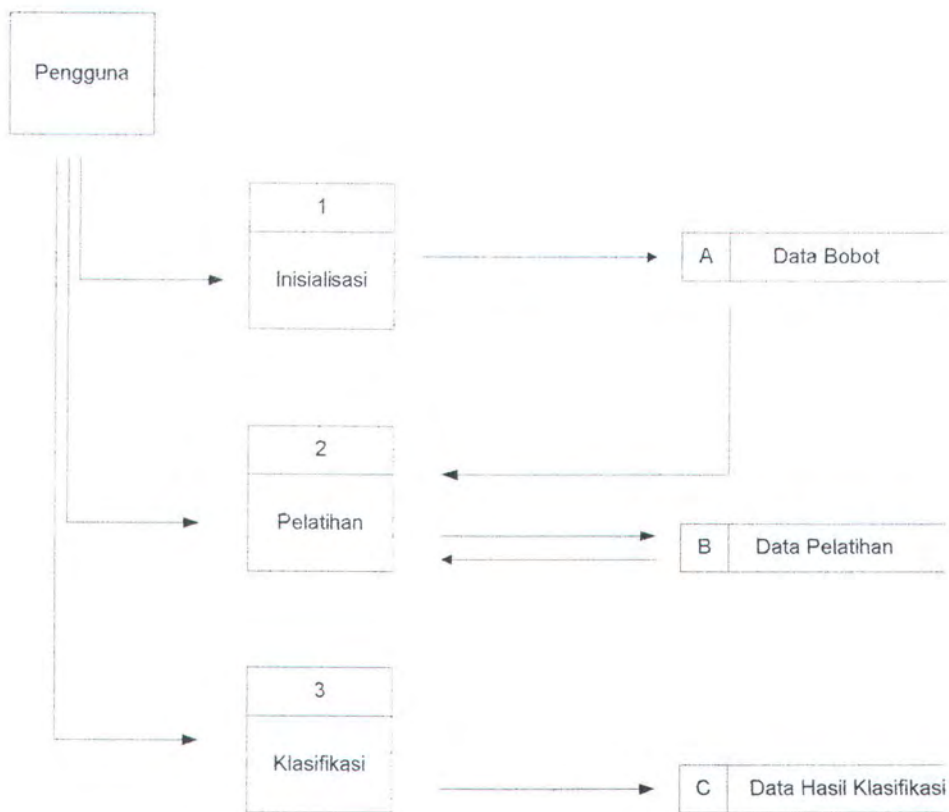
Gambar 4.6 DFD Level 0

Keterangan :

- Proses klasifikasi korosi akan mengolah data bobot sesuai dengan data pelatihan yang dipakai. Data bobot diolah pada proses pelatihan yang selanjutnya digunakan untuk proses klasifikasi.
- Data bobot, merupakan data bobot hasil inisialisasi atau pelatihan. Data ini disimpan dalam bentuk file teks dengan nama OW\*.CDBK, OrianW\*.CDBK, OWP\*.CDBK, atau OrianWP\*.CDBK.
- Data pelatihan, merupakan data yang digunakan untuk mengolah data bobot, berasal dari citra masukan. Data ini disimpan dalam bentuk file teks dengan nama \*.DAT
- Data hasil, yaitu data hasil proses klasifikasi, menyatakan kelas untuk setiap data masukan. Data ini disimpan dalam bentuk file teks, dengan nama \*.OUT.



Proses Klasifikasi citra korosi tersebut terdiri atas beberapa proses lagi seperti yang ditunjukkan oleh DFD level 1 pada gambar 4.7.



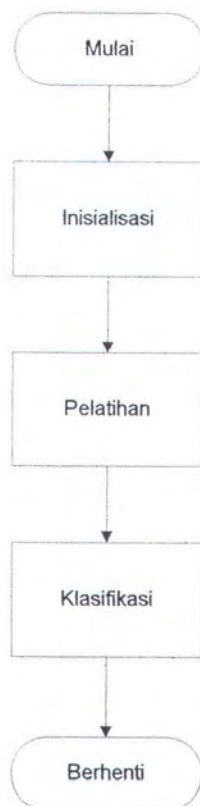
Gambar 4.7 DFD Level 1

Keterangan :

- Proses inisialisasi didahului oleh dekomposisi file masukan, sedangkan proses pelatihan dan klasifikasi bisa digunakan tanpa melalui proses dekomposisi, tergantung pada jenis masukan yang dipakai.
- Pelatihan dilakukan dengan menggunakan algoritma *LVQ*. Pelatihan dapat digunakan dengan menggunakan beberapa data sekaligus sampai sejumlah konstanta yang ditentukan.

- Klasifikasi seperti halnya pelatihan dapat digunakan untuk mengklasifikasikan beberapa masukan secara bersamaan.

Tahap-tahap proses klasifikasi citra korosi secara detail dapat dijelaskan dengan menggunakan diagram alir seperti pada gambar 4.8.

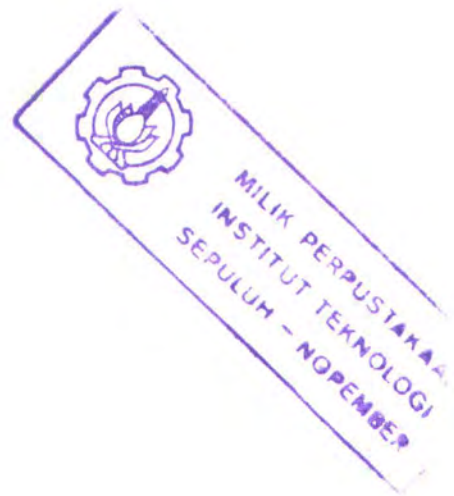
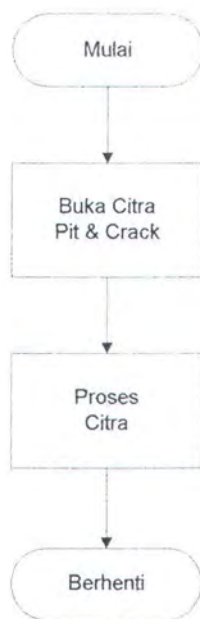


Gambar 4.8 Diagram Alir Proses

Proses inisialisasi merupakan tahapan untuk mendapatkan nilai bobot pertama kali dari satu citra korosi kelas *pit* dan satu citra korosi kelas *crack*. Proses pelatihan merupakan tahapan untuk merubah nilai bobot, disesuaikan dengan data pelatihan yang dipakai. Data pelatihan bisa berupa satu atau lebih citra. Proses



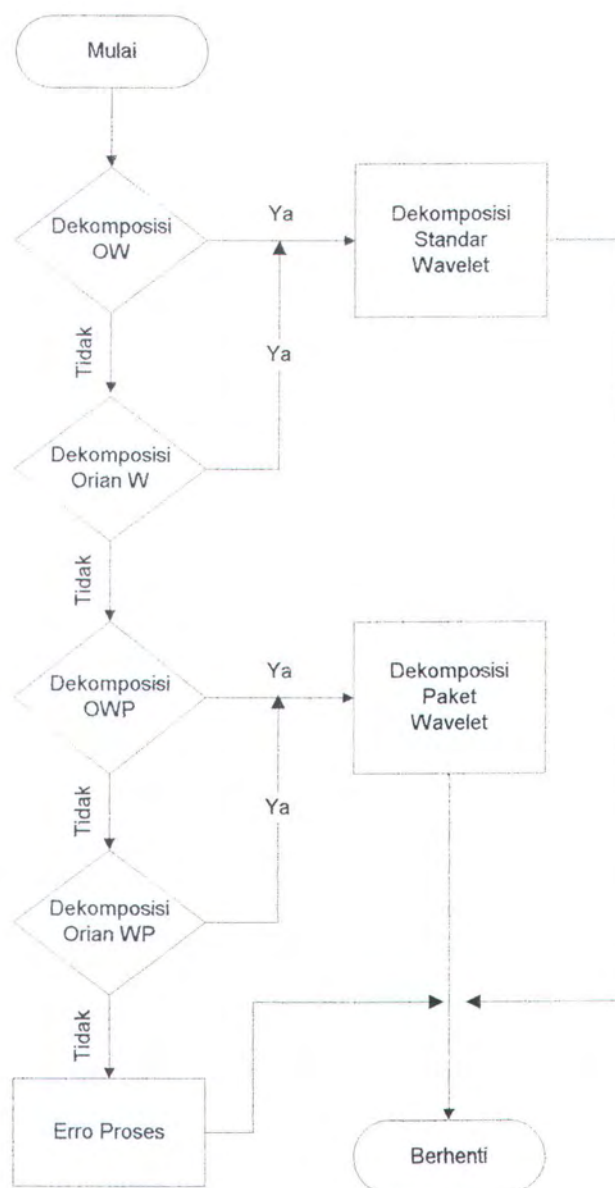
klasifikasi merupakan tahapan untuk mengklasifikasikan citra masukan ke dalam kelas yang ada berdasarkan nilai bobot yang diperoleh.



Gambar 4.9 Diagram Alir Proses Inisialisasi

Pada proses inisialisasi ini, digunakan modul proses citra secara terpisah. Hal ini dikarenakan proses tersebut juga akan digunakan pada proses pelatihan. Seperti yang telah dijelaskan pada bagian 4.2.3, modul proses citra merupakan modul untuk menentukan metode dekomposisi dan juga jenis feature yang akan dipakai. Metode dekomposisi dan cara pengambilan feature yang berbeda akan berakibat jumlah feature sebagai masukan untuk jaringan syaraf berbeda pula.

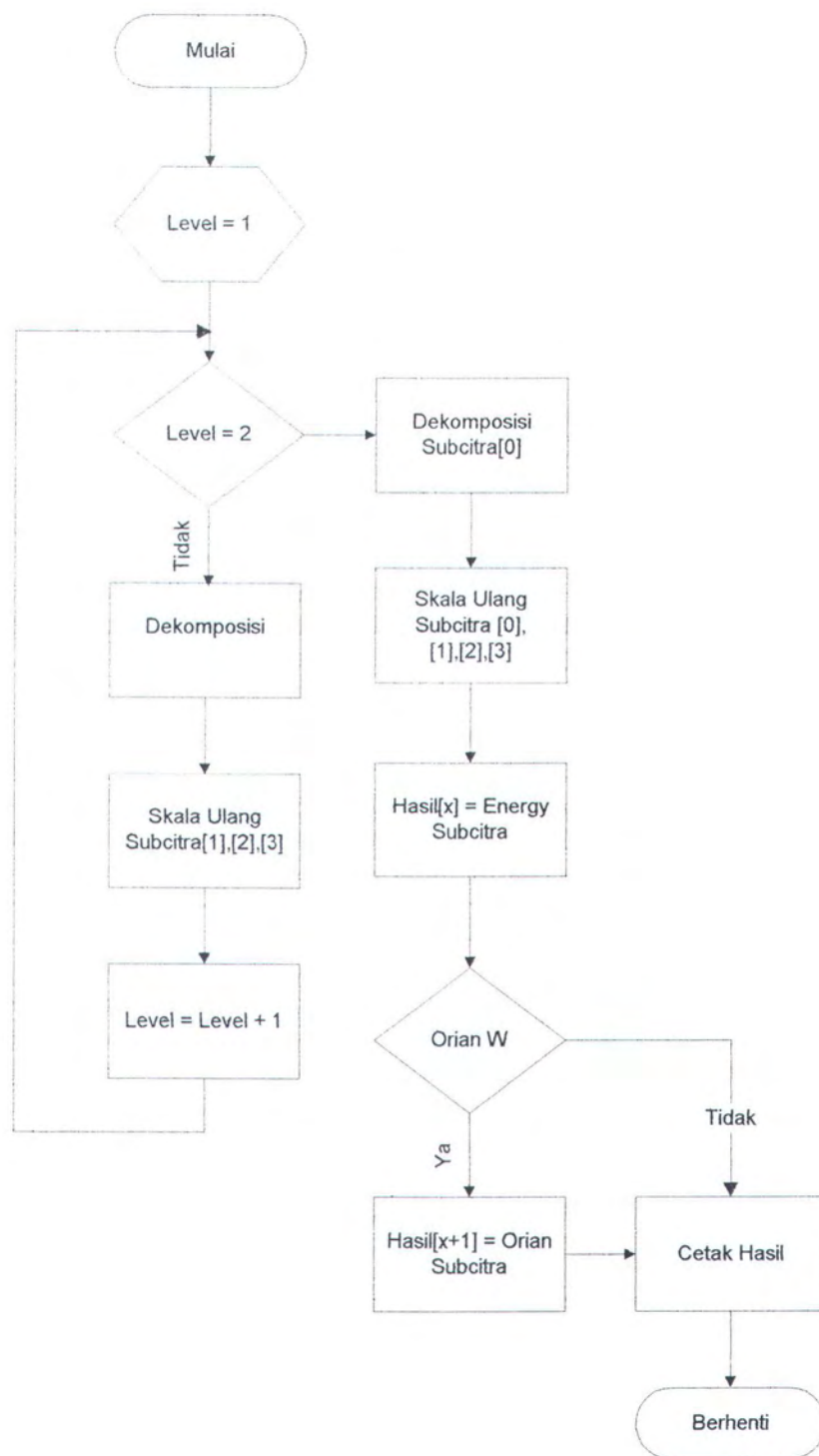
Langkah-langkah untuk melakukan pemrosesan citra tersebut dapat dijelaskan melalui gambar 4.10.



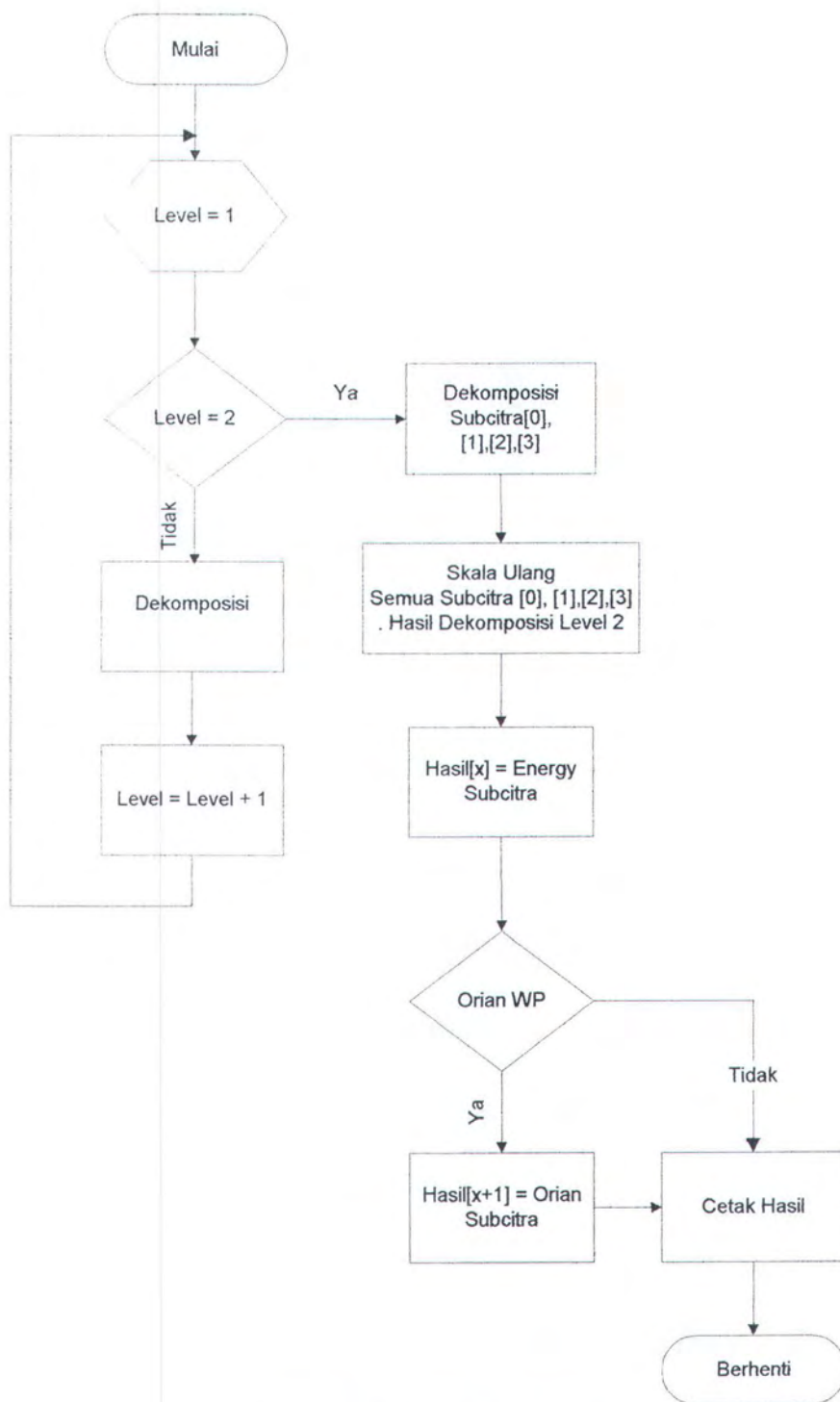
Gambar 4.10 Proses Citra

Metode dekomposisi dengan menggunakan wavelet standar dan wavelet paket pada dasarnya adalah sama. Perbedaannya akan muncul jika dekomposisi dilakukan sampai dengan level dua atau lebih. Wavelet paket akan menghasilkan subcitra yang lebih banyak daripada subcitra yang dihasilkan metode wavelet standar.





Gambar 4.11 Wavelet standar



Gambar 4.12 Wavelet paket

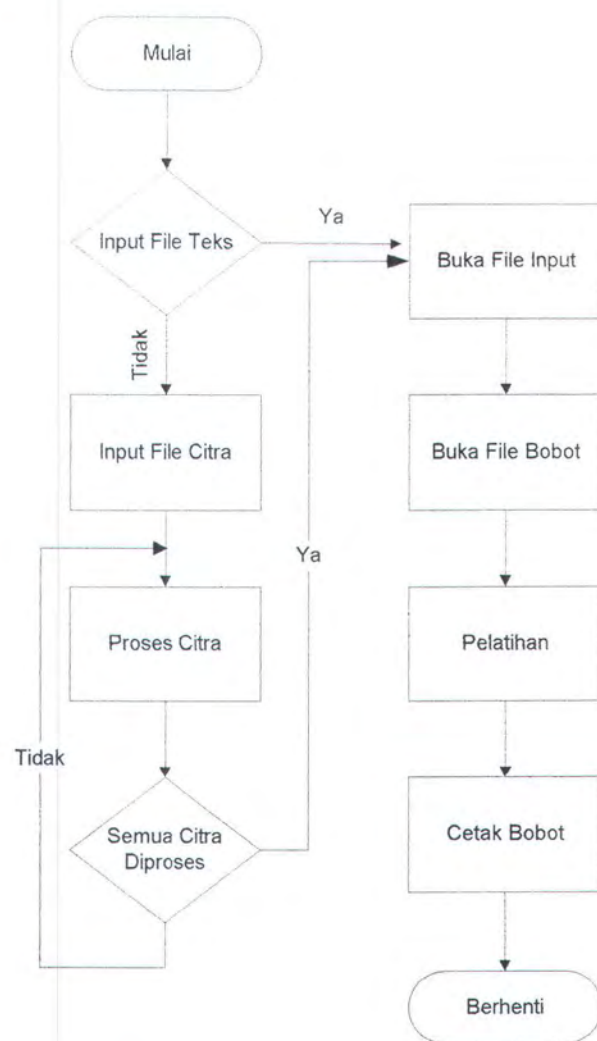
Proses pelatihan dapat dilakukan dengan menggunakan dua jenis masukan data yang berbeda, yaitu : file teks dan file citra. Jika data masukan berupa file teks,



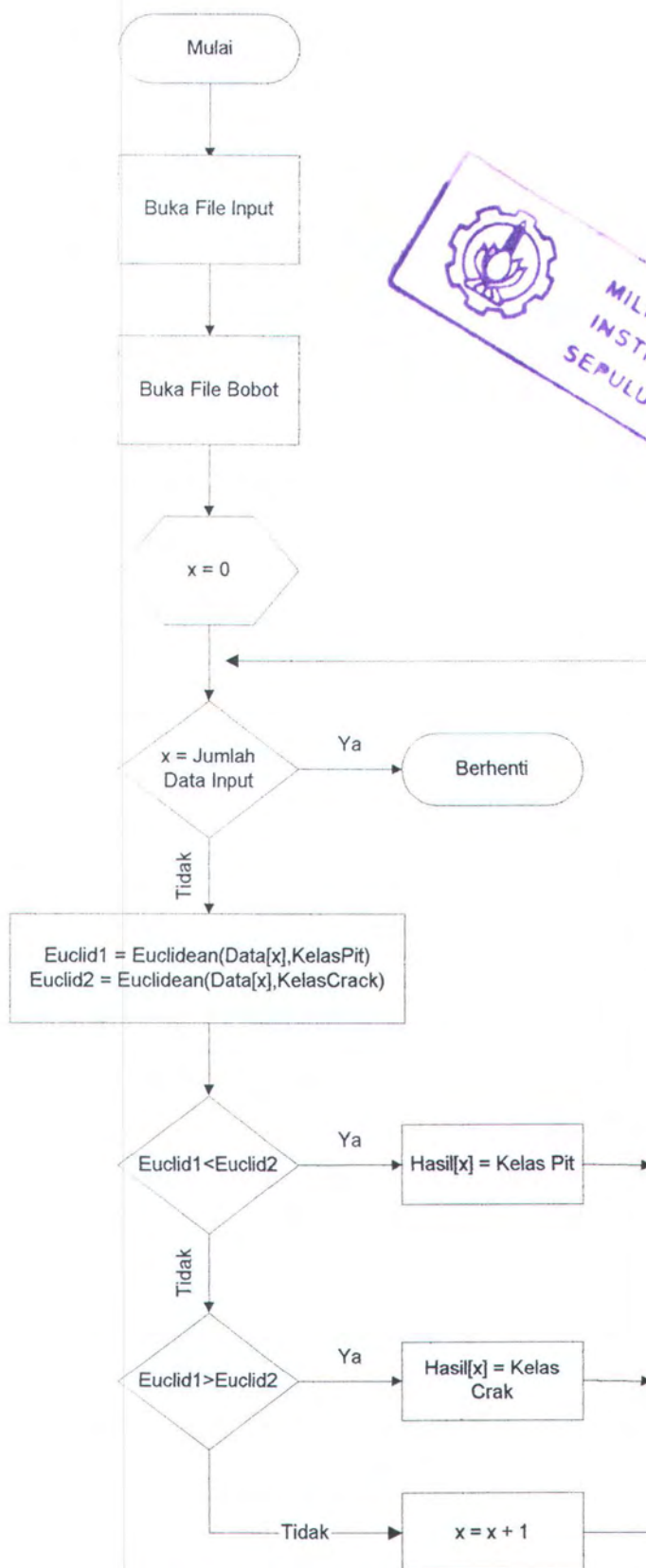
data akan langsung dibaca untuk proses pelatihan. Jika data masukan berupa file citra, akan dilakukan dekomposisi terlebih dahulu untuk mendapatkan nilai feature yang selanjutnya digunakan sebagai pelatihan, seperti pada diagram alir seperti pada gambar 4.13.

Modul pelatihan pada proses pelatihan di atas menggunakan algoritma *LVQ* seperti yang terdapat pada bab III.

Klasifikasi merupakan tahap akhir dalam perangkat lunak ini. Tahap ini akan menghasilkan klasifikasi terhadap citra masukan sesuai dengan bobot yang didapatkan dari proses pelatihan.



Gambar 4.13 Proses pelatihan

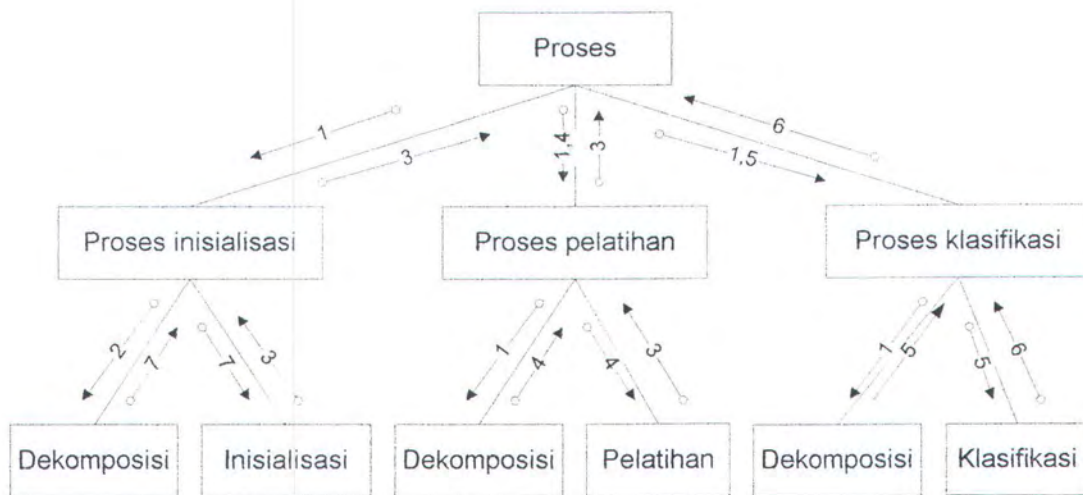


Gambar 4.10 Proses klasifikasi



#### 4.2.4 Hirarki Modul

Hirarki modul dari sistem klasifikasi citra korosi ini dapat dinyatakan sebagai berikut (gambar 4.15).



Gambar 4.15 Hirarki modul sistem klasifikasi

Pada gambar hirarki modul di atas terdapat angka-angka yang menunjukkan data yang digunakan dan dihasilkan oleh proses. Penjelasan angka-angka tersebut adalah sebagai berikut :

1. Citra input
2. Data piksel
3. Data bobot
4. Data pelatihan
5. Data klasifikasi
6. Vektor hasil klasifikasi
7. Data energy dan orian

### 4.3 Pembuatan Perangkat Lunak

#### 4.3.1 Implementasi Struktur Data

Struktur data yang digunakan dalam perangkat lunak ini dimaksudkan untuk mempermudah pembuatan dan penggunaannya sesuai dengan keperluan yang ada.

Adapun struktur data yang digunakan adalah sebagai berikut :

```
type
  PSingle = ^single;
  Daftar = TList;
  ASingle = array of single;
  AASingle = array of ASingle;
  TGambar1D = array of Daftar;
  TGambar2D = array of TGambar1D;

  //untuk menyimpan data file yang digunakan untuk pelatihan
  TDataTrain = record
    namafile : string;
    kelas    : string;
  end;
  TDT = array of TDataTrain;

  // untuk melakukan pengolahan data sehingga didapatkan nilai
  //feature
  TProcess = class(TPagesDlg)
  protected
    procedure dekomposisi(var gambar_real, gambar_0,
      gambar_1,gambar_2,gambar_3:Daftar;
      ganjil:boolean);
    procedure kuadrat(gambar:Daftar;var energy:Daftar);
    procedure skala_ulang(var data : Daftar; min, max:
      SmallInt);
  end;
  // merupakan turunan dari kelas TProcess untuk menentukan nilai
  // kelas dan orian
```



### 4.3.2 Implementasi Proses

Bagian ini berisikan implementasi proses yang dipakai sesuai dengan perancangan yang telah dilakukan sebelumnya. Proses-proses tersebut diimplementasikan dalam bentuk *procedure* atau *function*, sesuai dengan kemudahan pembuatan dan penggunaannya. Seperti telah dijelaskan pada awal bab IV, proses utama dalam sistem ini tersusun atas tiga bagian utama, yaitu : inisialisasi, pelatihan dan klasifikasi.

#### 4.3.2.1 Proses Inisialisasi

Proses inisialisasi dilakukan terlebih dahulu dengan menentukan jenis dekomposisi dan jenis feature yang dipakai. Keluaran dari proses ini adalah nilai feature yang dipakai sebagai bobot awal.

Sebelum dilakukan proses dekomposisi, citra masukan akan dikonversi terlebih dahulu menjadi vektor, dengan prosedur sebagai berikut :

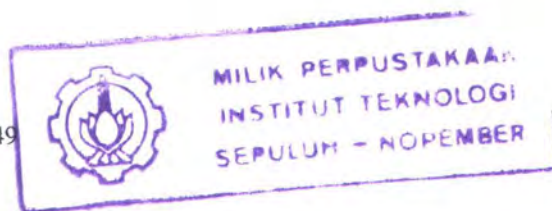
```
Procedure bmp2real(gambar_bmp : TBitmap;var gambar_real : Daftar);
```

Dengan *gambar\_bmp* merupakan citra masukan, dan *gambar\_real* merupakan nilai piksel yang akan diproses lebih lanjut.

Selanjutnya, dekomposisi dilakukan dengan prosedur berikut :

```
procedure TPagesDlg.paket_wavelet(pake_orian : boolean; gambar_real :  
    Daftar;proses,pit : boolean; nama : string);
```

```
procedure TPagesDlg.standar_wavelet(orian_pake : boolean;  
    gambar_real : Daftar; proses,pit : boolean; nama :  
    string);
```



Parameter yang digunakan pada dua prosedur tersebut adalah sebagai berikut :

- Variabel 'pake\_orian', dan 'orian\_pake' bernilai *true* jika digunakan metode orian W atau orian WP, dan bernilai *false* jika digunakan metode OW atau OWP.
- 'Gambar\_real' merupakan data yang akan didekomposisi.
- 'Proses' akan bernilai *true* jika terjadi pembuatan file teks baru, dan bernilai *false* jika data ditambahkan pada file teks yang sudah ada.
- 'Pit' bernilai *true* jika kelas data merupakan kelas *pit* dan bernilai *false* jika kelas data bukan *pit*.
- 'Nama' merupakan nama file teks yang akan dibuat atau ditambahi nilainya.

Data yang dihasilkan dari proses dekomposisi dihitung energi dan oriannya jika diperlukan, dengan fungsi berikut :

```
function hitung_energy_tot(var gambar : Daftar):single;
```

*gambar* merupakan data energi atau orian yang telah dihasilkan dari perhitungan subcitra-subcitra sebelumnya dengan persamaan pada bab II. Fungsi tersebut akan memberikan nilai balik berupa nilai tiap feature bertipe *single*.

#### 4.3.2.2 Proses Pelatihan

Proses pelatihan dilakukan dengan menggunakan algoritma *LVQ* seperti dijelaskan pada bab III. Masukan untuk proses pelatihan ini adalah file teks atau file citra. Jika masukan berupa file citra, file tersebut akan diproses terlebih dahulu dengan dekomposisi dan perhitungan energi baru dilakukan proses pelatihan seperti halnya masukan berupa file teks.



Proses pelatihan dilakukan dengan sejumlah iterasi tertentu yang merupakan masukan dari pengguna. Nilai alpha ditentukan  $0 < \alpha < 1$  dan untuk setiap iterasi dilakukan pengurangan nilai alpha dengan persamaan :

$$\alpha := \alpha * (1 - t / \text{steps})$$

dengan :

- *Steps* merupakan jumlah iterasi yang akan dilakukan
- *t* merupakan jumlah iterasi yang sudah dilakukan ditambah satu
- iterasi akan berhenti jika  $t = \text{steps}$ .

Algoritma *LVQ* diimplementasikan dengan program sebagai berikut :

```
// LAKUKAN TRAINING JIKA ADA DATA
if jumlah_input > 0 then
  while stop = false do
    begin
      inc(t);
      for x := 1 to jumlah_input do
        begin
          euclid1 := euclidean(dim,code_train[x],codebook[1]);
          euclid2 := euclidean(dim,code_train[x],codebook[2]);
          if (euclid1 < euclid2) then
            pit := true   else { KELAS PIT }
          if (euclid1 > euclid2) then pit := false; // KELAS CRACK

          // UPDATE CODEBOOK KELAS PIT
          // EUCLID = KELAS AKTUAL = PIT
          // JIKA EUCLID 1 = EUCLID 2 MAKA TIDAK ADA PERUBAHAN
          if (euclid1 < euclid2) or (euclid1 > euclid2) then
            begin
              if (pit = true ) and (code_train[x,0] = 1) then
                begin
                  for y := 0 to (dim -1) do
                    codebook[1,y] := codebook[1,y] + alpha *
                                          (code_train[x,y+1]-codebook[1,y]);
                end
            end
        end
      end
    end
  end
```

#### 4.3.2.3 Proses Klasifikasi

Proses klasifikasi dilakukan seperti halnya proses pelatihan, yaitu data dapat berupa file teks atau file citra. Jika data masukan berupa file citra, maka proses klasifikasi akan membuat file teks berdasarkan file citra tersebut.

Penentuan kelas citra dilakukan dengan menghitung nilai euclidian untuk setiap nilai feature dengan data bobot.

```
euclid1 := euclidean(dim,code_tes[x],codebook[1]);  
euclid2 := euclidean(dim,code_tes[x],codebook[2]);  
if (euclid1) < (euclid2 ) then  
    hasil[x] := 1 { KELAS PIT } else  
if (euclid1) > (euclid2 ) then hasil[x] := 2; // KELAS CRACK
```

Nilai Euclidian terkecil merupakan kelas yang didapatkan oleh data masukan.

#### 4.4 Tampilan Antarmuka

Tampilan antarmuka ini dibuat dengan menggunakan fasilitas *tab* pada bahasa pemrograman Borland Delphi 4.0. Dengan menggunakan pemrograman berbasis GUI, pengguna akan lebih mudah untuk memahami cara menjalankan sistem klasifikasi ini. Informasi dari sistem ini terdiri atas 4 bagian, yaitu :

- Inisialisasi
- Pelatihan
- Klasifikasi
- About

Informasi-informasi di atas dapat diperoleh dengan mengaktifkan salah satu tab pilihan pada program. Dari semua informasi di atas, terdapat bagian-bagian yang



harus diisi oleh pengguna. Untuk menghindari kesalahan urutan pengisiannya, satu bagian akan aktif setelah bagian masukan sebelumnya diisi.

#### 4.4.1 Inisialisasi

Pada bagian inisialisasi ini, pengguna harus mengisi citra untuk kelas *pit* dan *crack* sebagai nilai bobot awal. Di sini, harus dipilih jumlah *feature* yang akan digunakan, dengan memilih *radio button* yang tersedia. Hasil proses inisialisasi ini berupa nilai bobot (*codebook*), yang digunakan untuk proses pelatihan atau untuk proses klasifikasi. File keluaran proses inisialisasi ini secara langsung akan berekstensi ‘.CDBK’. Proses inisialisasi ditunjukkan oleh gambar 4.16.

#### 4.4.2 Pelatihan

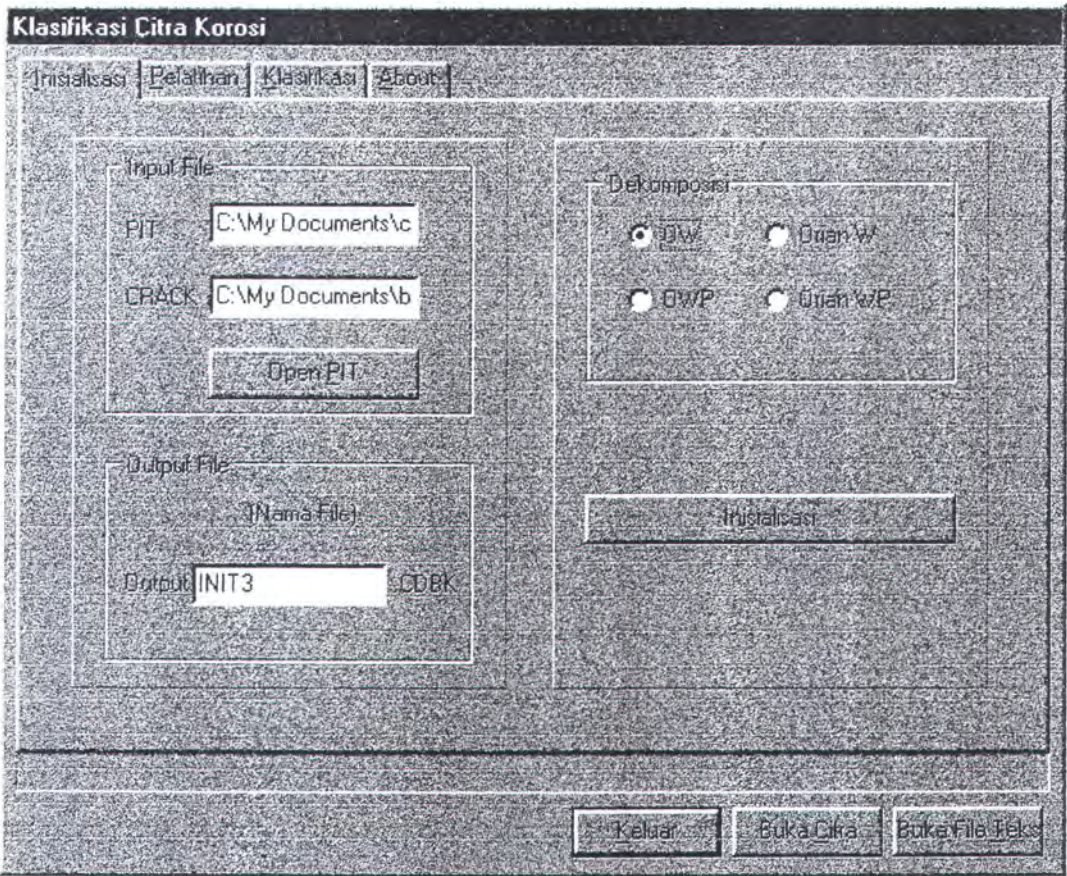
File keluaran dari proses inisialisasi atau nilai bobot yang sudah ada, digunakan sebagai data masukan yang akan dilatih (*training*). Data yang digunakan untuk pelatihan tersebut bisa berupa citra atau file teks. Seperti pada proses inisialisasi, proses pelatihan ini akan menghasilkan file bobot yang berekstensi ‘.CDBK’, untuk menandakan bahwa file tersebut merupakan file bobot (*codebook*). Proses pelatihan ditunjukkan oleh gambar 4.17.

#### 4.4.3 Klasifikasi

Proses klasifikasi akan menghasilkan file keluaran berupa file berekstensi ‘.OUT’. Sehingga hasil proses klasifikasi ini tidak akan ditampilkan secara langsung, tetapi perangkat lunak ini juga menyediakan fasilitas untuk membuka setiap file teks

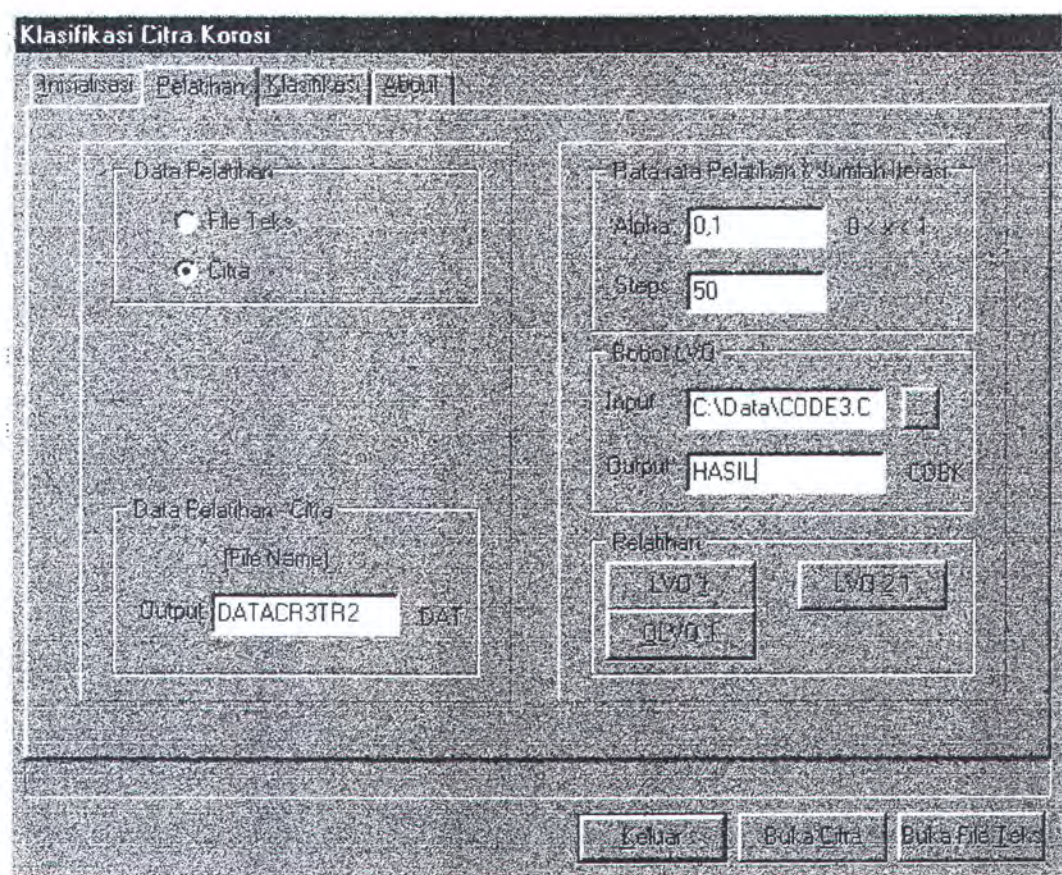


yang berhubungan dengan sistem ini, yaitu file yang berekstensi “.CDBK”, “.DAT”, dan “.OUT”, selain fasilitas untuk menampilkan citra.

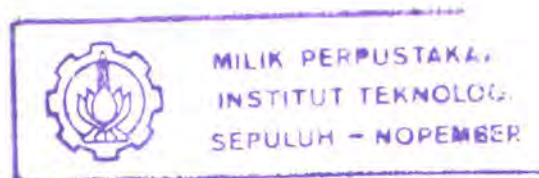


Gambar 4.16 Tampilan inisialisasi

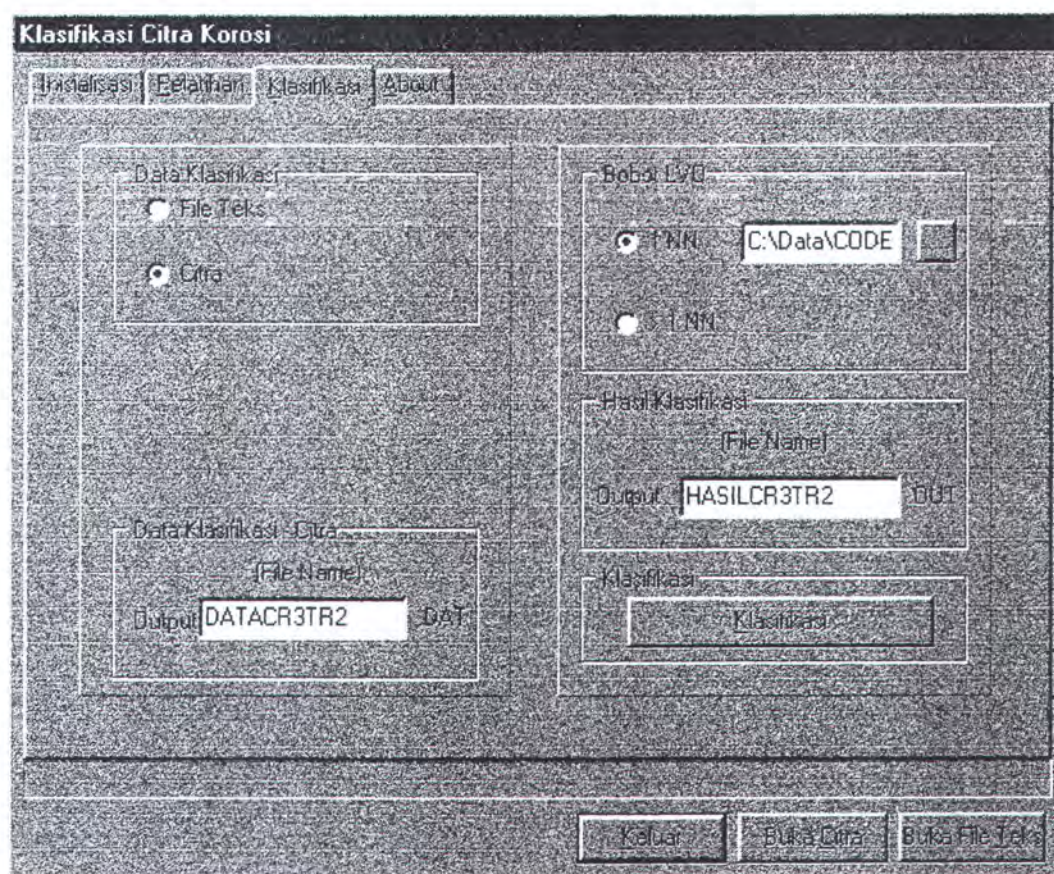




Gambar 4.17 Proses pelatihan







Gambar 4.18 Proses klasifikasi





## **BAB V**

# **UJI COBA DAN PEMBAHASANNYA**



## BAB V

### UJI COBA DAN PEMBAHASANNYA

#### 5.1 Uji Coba dan Analisis Hasil Klasifikasi

Pada bab ini akan dibahas hasil percobaan dari perangkat lunak yang dibuat. Pengujian perangkat lunak ini dilakukan dengan menggunakan jumlah data pelatihan yang berbeda, dan beberapa metode pelatihan yang berbeda pula, yaitu : *learning vector quantization* 1 (LVQ1), LVQ2.1, OLVQ1, untuk mengetahui kemampuan klasifikasi.

Pengujian dilakukan setelah dilakukan proses pelatihan. Terdapat dua macam pelatihan yang dipakai berdasarkan jumlah data yang digunakan. Masing-masing pelatihan tersebut dilakukan dengan tiga metode jaringan syaraf yang berbeda.

Klasifikasi dilakukan terhadap 5 kelompok kelas *pit* dan 5 kelompok kelas *crack*. Masing-masing kelompok terdiri atas 15 citra yang berbeda. Pengujian dilakukan dengan beberapa metode, untuk mendapatkan metode terbaik yang memberikan tingkat keberhasilan klasifikasi tertinggi. Pengujian tersebut dilakukan dengan :

- 14 data pelatihan
- 28 data pelatihan
- kombinasi dimensi yang sama
- 3 bobot terbaik
- citra yang dicerminkan terhadap garis horizontal
- citra yang berisi korosi kelas pit dan crack



Seperti yang telah dijelaskan pada bab IV, bahwa hasil klasifikasi yang dilakukan disimpan dalam file teks berekstensi '.OUT', untuk memudahkan pembacaannya. Contoh format file hasil klasifikasi adalah seperti pada gambar 5.1. sedangkan contoh citra yang diuji sesuai dengan kelas masing-masing adalah seperti pada gambar 5.2 dan 5.3

dimens

kelas yang benar

kelas hasil klasifikasi

kesalahan klasifikasi

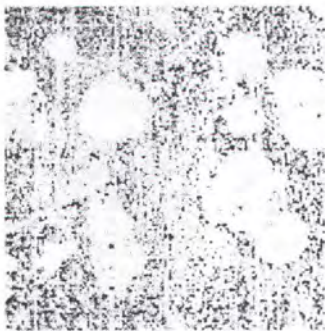
file yang diklasifikasikan

target = crack	klasifikasi = crack	C:\Data\gambar\lc\2\cr172764.bmp5
target = crack	klasifikasi = crack	C:\Data\gambar\lc\2\cr172818.bmp5
target = crack	klasifikasi = crack	C:\Data\gambar\lc\2\cr18289.bmp5
target = crack	klasifikasi = crack	C:\Data\gambar\lc\2\cr18304a.bmp5
target = crack	klasifikasi = crack	C:\Data\gambar\lc\2\cr18305.bmp5
target = crack	klasifikasi = crack	C:\Data\gambar\lc\2\cr18306.bmp5
target = crack	klasifikasi = crack	C:\Data\gambar\lc\2\cr19358a.bmp5
target = crack	klasifikasi = crack	C:\Data\gambar\lc\2\cr19358b.bmp5
target = crack	klasifikasi = crack	C:\Data\gambar\lc\2\cr19351b.bmp5
target = crack	klasifikasi = crack	C:\Data\gambar\lc\2\cr19352a.bmp5
target = crack	klasifikasi = crack	C:\Data\gambar\lc\2\cr20193.bmp5
target = crack	klasifikasi = pit	C:\Data\gambar\lc\2\cr22115b.bmp5
target = crack	klasifikasi = crack	C:\Data\gambar\lc\2\cr22126.bmp5
target = crack	klasifikasi = crack	C:\Data\gambar\lc\2\cr22185.bmp5
target = crack	klasifikasi = crack	C:\Data\gambar\lc\2\cr27174a.bmp5

Gambar 5.1 Format file hasil klasifikasi



Gambar 5.2 Citra korosi kelas crack



Gambar 5.3 Citra korosi kelas pit

### 5.1.1 Pengujian Data Menggunakan 14 Citra Pelatihan

Pada jaringan syaraf dilakukan pelatihan menggunakan 7 citra kelas *pit* dan 7 citra kelas *crack*, dengan nilai *alpha* 0,1 dan *steps* 50. Klasifikasi dilakukan dengan 3 metode pelatihan jaringan syaraf yang berbeda, yaitu LVQ1, LVQ2.1 dan OLVQ1. Dari ketiga pelatihan tersebut akan didapatkan bobot dengan 3, 4, 5 dan 8 jumlah feature (dimensi) per metode yang berbeda, sehingga didapatkan 12 buah bobot. bobot tersebut adalah LVQ1 dengan 3, 4, 5 dan 8 dimensi; LVQ2.1 dengan 3, 4, 5 dan 8 dimensi; OLVQ1 dengan 3, 4, 5 dan 8 dimensi.

Setiap bobot yang mempunyai dimensi sama dimasukkan ke dalam satu kelompok. Dari 12 buah bobot didapatkan 4 kelompok, dan setiap kelompok terdiri atas 3 buah bobot. Setiap kelompok digunakan untuk mengklasifikasikan citra yang diuji. Sehingga setiap citra dilakukan 3 kali perhitungan. Suara terbanyak dari 3 perhitungan tersebut akan menentukan jenis kelas citra tersebut, apakah termasuk kelas *pit* atau *crack*.

Dari pengujian citra menggunakan 14 citra pelatihan ini didapatkan tingkat keberhasilan klasifikasi tertinggi sebesar 80%, yaitu untuk pelatihan menggunakan LVQ2.1 dengan dimensi 3 (no 5) dan 4 (no 7). Hasil pengujian yang dilakukan terhadap 150 citra terdapat pada tabel 5.1.

### 5.1.2 Pengujian Data Menggunakan 28 Citra Pelatihan

Pada dasarnya pengujian dengan metode ini sama dengan metode pada bab 5.1.1. Perbedaannya adalah bobot yang dipakai untuk pengujian. Bobot yang dipakai pada bagian ini adalah hasil pelatihan menggunakan 14 citra *pit* dan 14 citra *crack*.





Dari pengujian ini didapatkan tingkat keberhasilan klasifikasi tertinggi sebesar 80,6667%, yaitu untuk pelatihan dengan menggunakan LVQ2.1 dengan dimensi 3 (no 21) dan 5 (no 22).

### **5.1.3 Pengujian Data Menggunakan 3 Codebook Terbaik**

Dari keseluruhan pengujian yang dilakukan, seperti yang dinyatakan pada tabel 5.1, didapatkan bahwa 3 bobot terbaik yang dapat menghasilkan persentase keberhasilan tertinggi adalah no 21, 22, 5 atau no 21, 22, 7; karena no 5 dan 7 mempunyai tingkat keberhasilan yang sama, yaitu 80%.

Dua kelompok codebook tersebut digunakan lagi untuk mengadakan pengujian agar didapatkan hasil yang lebih baik. Setiap kelompok ditentukan 3 bobot terbaik agar didapatkan hasil klasifikasi yang pasti, yaitu kelas pit atau crack.

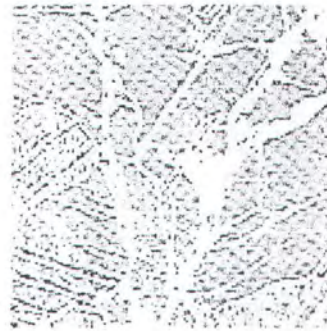
Seperti terdapat pada tabel 5.1, penggunaan dua kelompok bobot terbaik tersebut tidak dapat meningkatkan persentase keberhasilan klasifikasi.

### **5.1.4 Pengujian Data dengan Citra Dicerminkan Terhadap Garis Horizontal**

Proses klasifikasi terhadap suatu citra korosi kadang-kadang tidak dapat ditentukan arah atas-bawah atau kiri-kanannya, tergantung pada arah pengambilan gambar citra tersebut. Sehingga diperlukan uji coba terhadap citra yang arahnya berubah dari citra aslinya. Pada uji coba ini, citra asli dicerminkan terhadap garis horizontal pada tengah citra, sehingga didapatkan citra baru yang berbeda dari citra asli, seperti pada gambar 5.5.



Gambar 5.4 Citra awal



Gambar 5.5 Citra hasil pencerminan

Pengujian dilakukan terhadap 15 data masukan citra kelas *pit* dan 15 data masukan citra kelas *crack* yang telah dicerminkan terhadap garis horizontal menggunakan bobot dengan dimensi 3. Hasil pengujian menunjukkan bahwa pengubahan arah tidak mempengaruhi hasil klasifikasi. Dari pengujian tersebut didapatkan 3 kesalahan klasifikasi; dimana kesalahan tersebut sama persis dengan kesalahan klasifikasi yang dihasilkan oleh citra sebelum pencerminan.

#### 5.1.5 Pengujian Data Citra yang Berisi Korosi Kelas Pit dan Crack

Pada umumnya, pada satu citra korosi hanya terdapat satu tipe atau morfologi korosi, tetapi tidak menutup kemungkinan bahwa pada satu citra terdapat lebih dari satu tipe korosi. Untuk itu perlu dilakukan pengujian terhadap citra yang mengandung lebih dari satu tipe korosi.

Uji coba yang dilakukan pada bagian ini dilakukan dengan menggabungkan dua tipe korosi yang ada menjadi satu citra. Penggabungan citra dua tipe korosi tersebut dilakukan dalam sebelas tahap, yaitu 1%, 10%, 20%, . . . , 100%. Tahap 1% merupakan citra korosi kelas *pit*, sedangkan 100% merupakan citra korosi kelas *crack*. Citra gabungan atau kombinasi tersebut ditunjukkan pada gambar 5.6 - 5.16





Gambar 5.6  
Kombinasi 1%



Gambar 5.7  
Kombinasi 10%



Gambar 5.8  
Kombinasi 20%



Gambar 5.9  
Kombinasi 30%



Gambar 5.10  
Kombinasi 40%



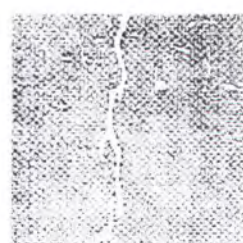
Gambar 5.11  
Kombinasi 50%



Gambar 5.12  
Kombinasi 60%



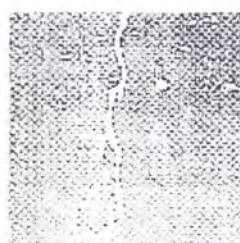
Gambar 5.13  
Kombinasi 70%



Gambar 5.14  
Kombinasi 80%



Gambar 5.15  
Kombinasi 90%



Gambar 5.16  
Kombinasi 100%

Dari uji coba yang dilakukan, citra kombinasi korosi dengan kapasitas 1% sampai 50% diklasifikasikan sebagai korosi kelas pit, sedangkan citra kombinasi korosi kapasitas 60% sampai dengan 100% diklasifikasikan sebagai korosi kelas crack.

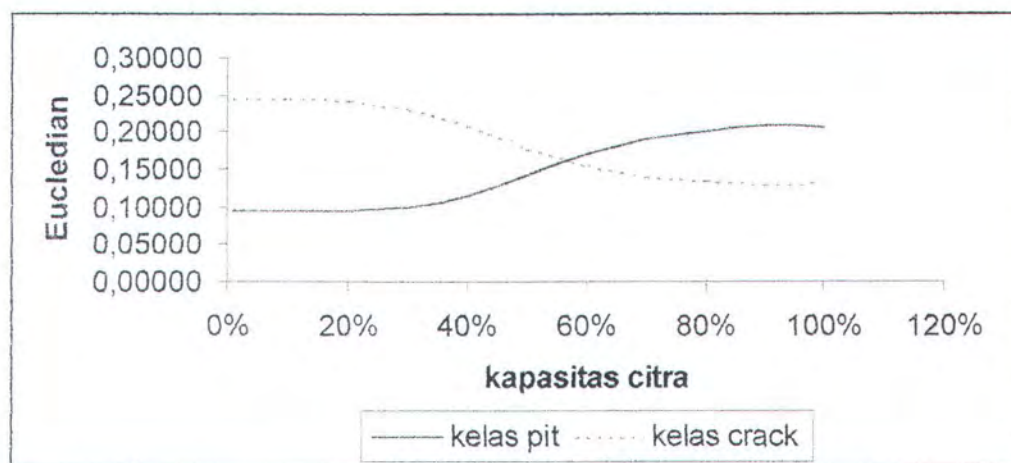
KAPASITAS	EUCLEDIAN		JARAK PIT DAN CRACK   2 - 3	HASIL KLASIFIKASI
	PIT	CRACK		
1	2	3	4	5
1%	0,09450	0,24359	0,14909	PIT
10%	0,09415	0,24416	0,15001	PIT
20%	0,09426	0,24267	0,14841	PIT
30%	0,09965	0,23192	0,13227	PIT
40%	0,11464	0,20903	0,09439	PIT
50%	0,14240	0,17801	0,03561	PIT
60%	0,16981	0,15459	0,01522	CRACK
70%	0,19029	0,14093	0,04936	CRACK
80%	0,20205	0,13384	0,06821	CRACK
90%	0,20739	0,13057	0,07682	CRACK
100%	0,20675	0,13159	0,07516	CRACK

Tabel 5.1 Klasifikasi citra kombinasi

Dari tabel 5.1 dapat diketahui bahwa citra kombinasi dengan kapasitas 1% sampai 50%, kelas pit merupakan korosi yang dominan, sedangkan untuk citra kombinasi dengan kapasitas 60% sampai dengan 100% kelas crack lebih dominan.

Selisih nilai Euclidian antara kelas pit dan crack akan semakin kecil sebanding dengan semakin 'imbang'nya tipe pit dan crack pada citra tersebut; dengan nilai tekecil dicapai pada saat kapasitas citra bernilai 60%. Pada kapasitas tersebut, baik kelas pit maupun crack relatif terlihat lebih jelas daripada citra dengan nilai kapasitas yang lain. Dalam bentuk kurva, tabel 5.1 dapat dinyatakan seperti pada gambar 5.17 berikut.





Gambar 5.17 Kurva nilai Eucledian

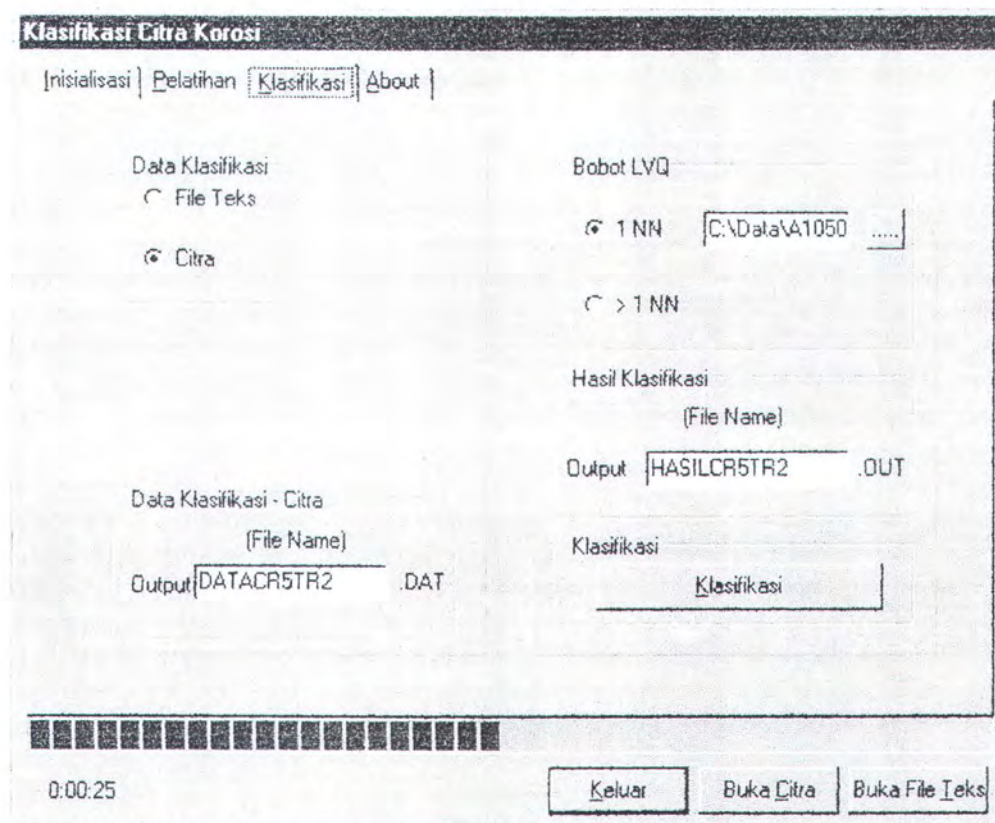
## 5.2 Uji Coba dan Analisis Kecepatan Proses Sistem

Proses klasifikasi dan pelatihan dilakukan dengan dua macam pilihan data masukan, yaitu file teks atau citra. Pada komputer yang sama, dua macam data masukan ini akan memerlukan waktu proses yang berbeda. Perangkat keras yang digunakan pada penelitian ini adalah PC Pentium 100 dengan RAM 32 Mbyte dan *Graphics Card* 1 Mbyte dan *Chace memory* 0 Mbyte. Contoh proses klasifikasi terdapat pada gambar 5.18.

Dengan data masukan berupa file teks, proses klasifikasi atau pelatihan dapat langsung dilakukan sehingga proses dapat berlangsung jauh lebih cepat, yaitu rata-rata memerlukan 0,0 detik. Jika data masukan berupa citra, harus dilakukan dekomposisi sebelum dilakukan proses pelatihan atau klasifikasi. Proses klasifikasi terhadap 15 citra menggunakan 3 *feature* (dimensi 3) memerlukan waktu 55 detik; dimensi 5 memerlukan waktu 58 detik; dimensi 4 memerlukan waktu 1 menit 10 detik; dimensi 8 memerlukan waktu 1 menit 15 detik.

Dimensi 3 dan 5 memerlukan waktu yang lebih sedikit daripada dimensi 4 dan 8 karena proses dekomposisi untuk dimensi 3 dan 5 menggunakan wavelet standar sedangkan dimensi 4 dan 8 menggunakan wavelet paket.

Pada proses pelatihan dan klasifikasi ini, proses terlama terjadi pada saat dekomposisi citra, bukan terjadi pada pelatihan ataupun klasifikasi. Perhitungan waktu yang diperlukan untuk proses klasifikasi ditunjukkan pada gambar 5.18.



Gambar 5.18 Proses klasifikasi dan perhitungan waktunya



ALPHA 0,1  
 STEPS 50  
 PELATIHAN DENGAN 7 CITRA PIT DAN 7 CITRA CRACK

METODE JARINGAN SYARAF	NO	JUMLAH FEATURE	KELAS	KELOMPOK					JUMLAH KESALAHAN KLASIFIKASI	PERSENTASE KESALAHAN KLASIFIKASI	PERSENTASE KETEPATAN KLASIFIKASI
				1	2	3	4	5			
LVQ1	1	3	PIT	4	8	0	5	4	21	0,220000	0,780000
			CRACK	0	1	2	2	7	12		
	2	5	PIT	3	8	2	5	4	22	0,226667	0,773333
			CRACK	0	1	2	2	7	12		
	3	4	PIT	4	8	0	7	6	25	0,246667	0,753333
			CRACK	1	1	2	2	6	12		
	4	8	PIT	3	9	2	6	8	28	0,320000	0,680000
			CRACK	1	3	2	5	9	20		
LVQ2.1	5	3	PIT	2	5	0	1	3	11	0,200000	0,800000
			CRACK	2	1	3	5	8	19		
	6	5	PIT	1	7	2	5	5	20	0,206667	0,793333
			CRACK	1	1	0	1	8	11		
	7	4	PIT	1	6	0	1	2	10	0,200000	0,800000
			CRACK	3	1	3	5	8	20		
	8	8	PIT	3	3	2	1	7	16	0,520000	0,480000
			CRACK	10	12	12	14	14	62		
OLVQ1	9	3	PIT	4	8	0	7	5	24	0,240000	0,760000
			CRACK	0	1	2	2	7	12		
	10	5	PIT	4	8	2	6	5	25	0,240000	0,760000
			CRACK	0	1	2	1	7	11		
	11	4	PIT	4	10	0	7	6	27	0,253333	0,746667
			CRACK	0	1	2	2	6	11		
	12	8	PIT	5	10	4	8	9	36	0,360000	0,640000
			CRACK	0	2	3	5	8	18		

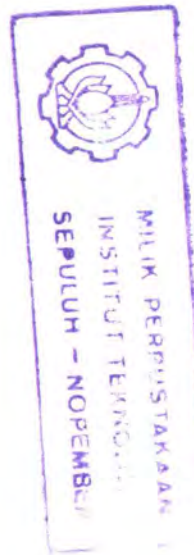


METODE JARINGAN SYARAF	NO	JUMLAH FEATURE	KELAS	KELOMPOK					JUMLAH KESALAHAN KLASIFIKASI	PERSENTASE KESALAHAN KLASIFIKASI	PERSENTASE KETEPATAN KLASIFIKASI
				1	2	3	4	5			
LVQ1,	13	3	PIT	4	8	0	5	4	21	0,220000	0,780000
			CRACK	0	1	2	2	7	12		
LVQ2.1,	14	5	PIT	3	8	2	5	4	22	0,220000	0,780000
			CRACK	0	1	2	1	7	11		
OLVQ1	15	4	PIT	4	8	0	7	6	25	0,246667	0,753333
			CRACK	1	1	2	2	6	12		
OLVQ1	16	8	PIT	3	9	2	6	9	29	0,326667	0,673333
			CRACK	0	3	3	5	9	20		

ALPHA 0,1  
 STEPS 50  
 PELATIHAN DENGAN 14 CITRA *PIT* DAN 14 CITRA *CRACK*

METODE JARINGAN SYARAF	NO	JUMLAH FEATURE	KELAS	KELOMPOK					JUMLAH KESALAHAN KLASIFIKASI	PERSENTASE KESALAHAN KLASIFIKASI	PERSENTASE KETEPATAN KLASIFIKASI
				1	2	3	4	5			
LVQ1	17	3	PIT	4	11	1	6	6	28	0,260000	0,740000
			CRACK	0	1	2	2	6	11		
	18	5	PIT	4	8	0	5	5	22	0,226667	0,773333
			CRACK	0	1	2	2	7	12		
	19	4	PIT	4	11	0	6	6	27	0,253333	0,746667
			CRACK	0	1	2	2	6	11		
	20	8	PIT	7	9	3	9	7	35	0,313333	0,686667
			CRACK	3	1	2	2	4	12		
LVQ2.1	21	3	PIT	1	5	0	1	2	9	0,193333	0,806667
			CRACK	2	1	3	6	8	20		
	22	5	PIT	3	8	0	2	4	17	0,193333	0,806667
			CRACK	0	1	2	2	7	12		
	23	4	PIT	2	6	0	1	3	12	0,213333	0,786667
			CRACK	3	1	3	5	8	20		
	24	8	PIT	7	11	5	11	10	44	0,320000	0,680000
			CRACK	0	0	0	1	3	4		





METODE JARINGAN SYARAF	NO	JUMLAH FEATURE	KELAS	KELOMPOK					JUMLAH KESALAHAN KLASIFIKASI	PERSENTASE KESALAHAN KLASIFIKASI	PERSENTASE KETEPATAN KLASIFIKASI
				1	2	3	4	5			
OLVQ1	25	3	PIT	4	10	1	5	6	26	0,246667	0,753333
			CRACK	0	1	2	2	6	11		
	26	5	PIT	4	8	0	5	5	22	0,226667	0,773333
			CRACK	0	1	2	2	7	12		
	27	4	PIT	4	11	0	5	6	26	0,246667	0,753333
			CRACK	0	1	2	2	6	11		
	28	8	PIT	5	8	2	6	6	27	0,260000	0,740000
			CRACK	3	1	2	2	4	12		
LVQ1, LVQ2.1, OLVQ1	29	3	PIT	4	10	1	5	6	26	0,246667	0,753333
			CRACK	0	1	2	2	6	11		
	30	5	PIT	4	8	0	5	5	22	0,226667	0,773333
			CRACK	0	1	2	2	7	12		
	31	4	PIT	4	11	0	5	6	26	0,246667	0,753333
			CRACK	0	1	2	2	6	11		
	32	8	PIT	6	9	3	9	7	34	0,306667	0,693333
			CRACK	3	1	2	2	4	12		

KLASIFIKASI MENGGUNAKAN KOMBINASI 3 CODEBOOK TERBAIK

METODE JARINGAN SYARAF	NO	JUMLAH FEATURE	KELAS	KELOMPOK					JUMLAH KESALAHAN KLASIFIKASI	PERSENTASE KESALAHAN KLASIFIKASI	PERSENTASE KETEPATAN KLASIFIKASI
				1	2	3	4	5			
(21),(22),(5)	33	3,5	PIT	2	5	0	1	3	11	0,200000	0,800000
			CRACK	2	1	3	5	8	19		
(21),(22),(7)	34	3,5,4	PIT	1	6	0	1	3	11	0,200000	0,800000
			CRACK	2	1	3	5	8	19		





## **BAB VI PENUTUP**



## BAB VI

### PENUTUP

#### 6.1 Kesimpulan

- Klasifikasi citra korosi biasanya dilakukan menggunakan *rule based system* dan sulit dilakukan dengan menggunakan pengolahan citra sederhana. Tetapi dapat juga diselesaikan dengan menggunakan metode analisis tekstur dan jaringan syaraf tiruan, yang dalam hal ini adalah *Learning Vector Quantization (LVQ)*.
- Penentuan banyaknya data untuk pelatihan harus dilakukan dengan teliti. Terlalu sedikit atau terlalu banyak data untuk pelatihan akan mengurangi ketepatan klasifikasi yang dilakukan. Banyaknya data pelatihan yang optimal dapat diketahui dengan melakukan uji coba terhadap data-data klasifikasi.
- Nilai *alpha* dan jumlah *steps* dipilih seoptimal mungkin dengan melakukan uji coba terhadap data.
- Penggunaan jumlah *feature* yang lebih banyak tidak selalu menghasilkan keluaran yang lebih baik. Berdasarkan waktu pemrosesan dan hasil klasifikasi yang dilakukan, penggunaan wavelet standar relatif lebih baik daripada wavelet paket. Sesuai dengan percobaan yang telah dilakukan, penggunaan wavelet standar (3 dan 5 dimensi) dengan pelatihan menggunakan LVQ2.1, nilai *alpha* = 0,1 dan *steps* = 50 merupakan metode terbaik. Dengan parameter-parameter tersebut didapatkan keakuratan sistem mencapai 80,6% untuk pengujian 150 data.

## 6.2 Saran

- Pengembangan sistem ini dapat dilakukan dengan menggunakannya secara *on line* terhubung dengan kamera, untuk melakukan klasifikasi pada daerah yang sulit dijangkau oleh manusia, misalnya di bagian kapal yang sangat sempit.
- Sistem ini terbatas pada data masukan berupa citra *grey scale*, sehingga untuk pengembangan berikutnya dapat dilakukan untuk citra berwarna.
- Nilai alpha dan steps serta jumlah data pelatihan dapat ditentukan lebih baik lagi dengan melakukan lebih banyak percobaan yang berbeda.
- Untuk meningkatkan kemampuan sistem ini, metode klasifikasi yang lain, seperti metode statistik, dapat dikombinasikan dengan LVQ. Sehingga akan didapatkan tingkat keberhasilan klasifikasi yang lebih baik.





## DAFTAR PUSTAKA



## DAFTAR PUSTAKA

1. Graps, Amara, *An Introduction to Wavelets, Signal & Image Processing*, IEEE Computation, Science & Engineering, 1995.
2. Mallat, Stephane G., *Multifrequency Channel Decomposition of Images and Wavelet Model*, IEEE Transaction on Acoustic, Speech, and Signal Processing, Vol. 37, No. 12, December 1989.
3. Mallat, Stephane G., *A Theory for Multiresolution Signal Decomposition : The Wavelet Representation*, IEEE Transaction on Pattern Analysis and Machine Intellegent, Vol. 11, No.7, July 1989.
4. Rafael, C. Gonzales and Woods, R.G., *Digital Image Processing*, Addison Wesley Publishing Company, Mass, 1992.
5. Deubechies, I, *Ten Lectures on Wavelet*, Rutgers University and AT&T Bell Laboratories, Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania, 1992.
6. Livens, Scheunders, Wouwer, Van Dyck, Smets, Winkelmanns, Bogaerts, *A Texture Analysis Approach to Corrosion Image Classification*, Microscopy, Microanalysis, Microstructure, 7(2), p.1 –10, April 1996.
7. Livens, Scheunders, Wouwer, Van Dyck, Smets, Winkelmanns, Bogaerts, *Classification of Corrosion Images by Wavelet Signatures and LVQ Network*, Proceedings Int. Conf. On Computer Analysis of Images and Patterns, LNCS 970, pp 538-543, Prague, Sept. 1995.
8. Livens, *Image Analysis for Material Characterization*, Universiteit Antwerpen, Antwerpen, 1998.
9. Schalkoff, Robert, *Pattern Recognition Statistical, Structural and Neural Approaches*, John Wiley & Sons, Inc., 1992.
10. Fausett, Laurene, *Fundamentals of Neural Networks Architectures, Algorithms, and Applications*, Prentice Hall International, 1994.
11. Davalo, Naim, *Neural Networks*, Macmillan Press Ltd., 1991.





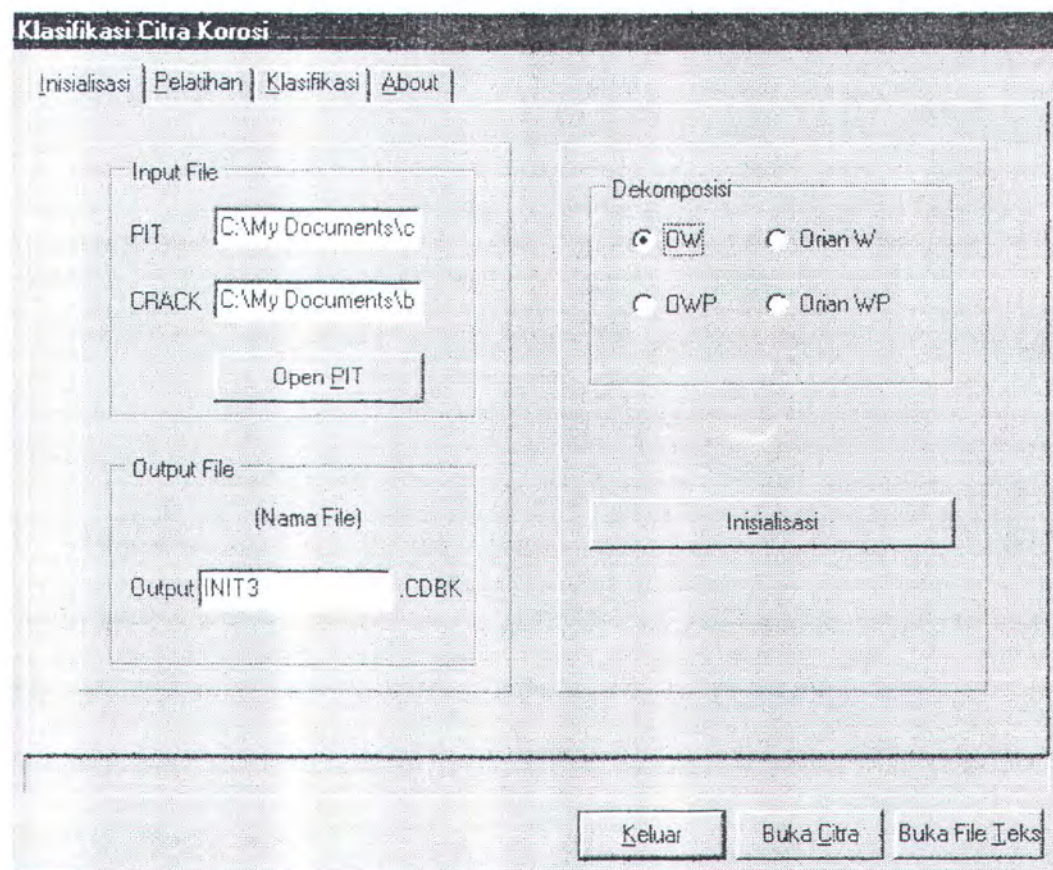
## LAMPIRAN



**LAMPIRAN**  
**PETUNJUK PENGGUNAAN**  
**PERANGKAT LUNAK**

Perangkat lunak ini dibagi dalam tiga tahap proses, yaitu inisialisasi, pelatihan dan klasifikasi. Dalam penggunaannya, harus tersedia direktori 'C:\Data', untuk menempatkan semua keluaran (*output*) program. Tujuan adanya pembatasan ini adalah untuk 'mengumpulkan' penempatan file keluaran tersebut dan juga untuk memudahkan pengisian data-data yang diperlukan oleh perangkat lunak.

**1. Inisialisasi**



Gambar Lampiran.1 Antarmuka proses inisialisasi



Inisialisasi merupakan proses untuk mendapatkan nilai awal bobot. Hasil proses inisialisasi dapat digunakan langsung untuk proses klasifikasi, dapat juga digunakan untuk proses pelatihan. Bobot tersebut disimpan dalam file teks dengan ekstensi '.CDBK'.

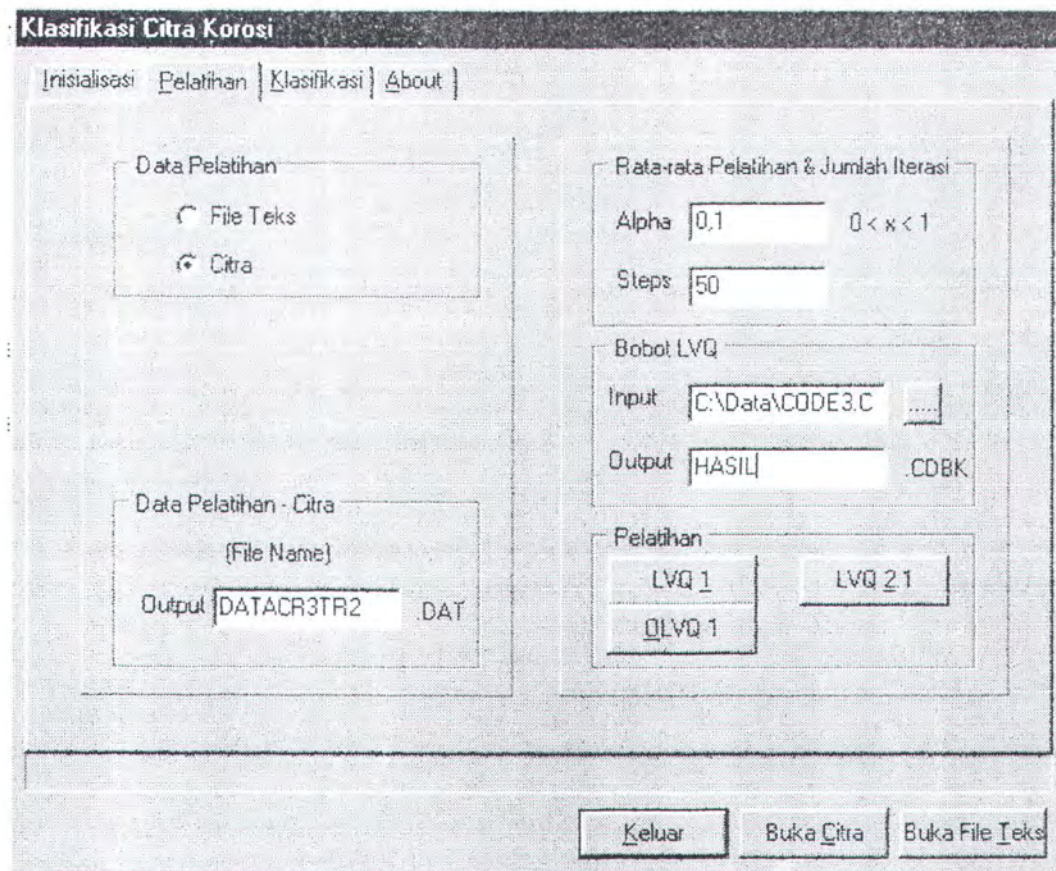
Inisialisasi dilakukan dengan memasukkan 1 citra korosi kelas pit dan 1 citra korosi kelas crack. Proses inisialisasi dilakukan dengan cara sebagai berikut :

- Pada bagian 'Input File', masukkan 1 citra kelas pit dan 1 citra kelas crack. Klik tombol 'Open pit' untuk memasukkan citra korosi pit. Setelah dimasukkan citra korosi kelas pit, tombol akan berubah menjadi 'Open crack'. Masukkan citra korosi kelas crack.
- Pada bagian 'Output File', masukkan nama file hasil proses inisialisasi tanpa ekstensi, karena perangkat lunak akan secara otomatis menambahkan ekstensi pada file tersebut, yaitu '\*.CDBK'.
- Pada bagian 'Dekomposisi', pilih salah satu metode yang dilakukan untuk proses dekomposisi.
  - OW menghasilkan keluaran 3 dimensi (wavelet standar)
  - OrianW menghasilkan keluaran 5 dimensi (wavelet standar)
  - WP menghasilkan keluaran 4 dimensi (wavelet paket)
  - Orian WP menghasilkan keluaran 8 dimensi (wavelet paket)
- Klik tombol 'Inisialisasi' untuk melakukan proses inisialisasi. Nama file teks hasil perangkat lunak ini akan berubah secara otomatis sesuai metode dekomposisi yang digunakan. Misalnya, jika diisikan nama xxx pada bagian 'Output File' dan digunakan metode OW, maka hasil proses inisialisasi akan



menghasilkan file : OWxxx.CDBK; jika digunakan metode Orian WP, maka akan dihasilkan file : OrianWP.CDBK.

## 2. Pelatihan



Gambar Lampiran.2 Antarmuka proses pelatihan

Proses pelatihan menggunakan data masukan berupa file teks atau file citra. Pemilihan jenis data masukan tersebut dilakukan dengan memilih salah satu *radio button* pada bagian data pelatihan. File teks yang digunakan untuk proses pelatihan ini diperoleh dari proses pelatihan atau klasifikasi sebelumnya yang menggunakan data masukan berupa citra.



Proses pelatihan dilakukan dengan cara sebagai berikut :

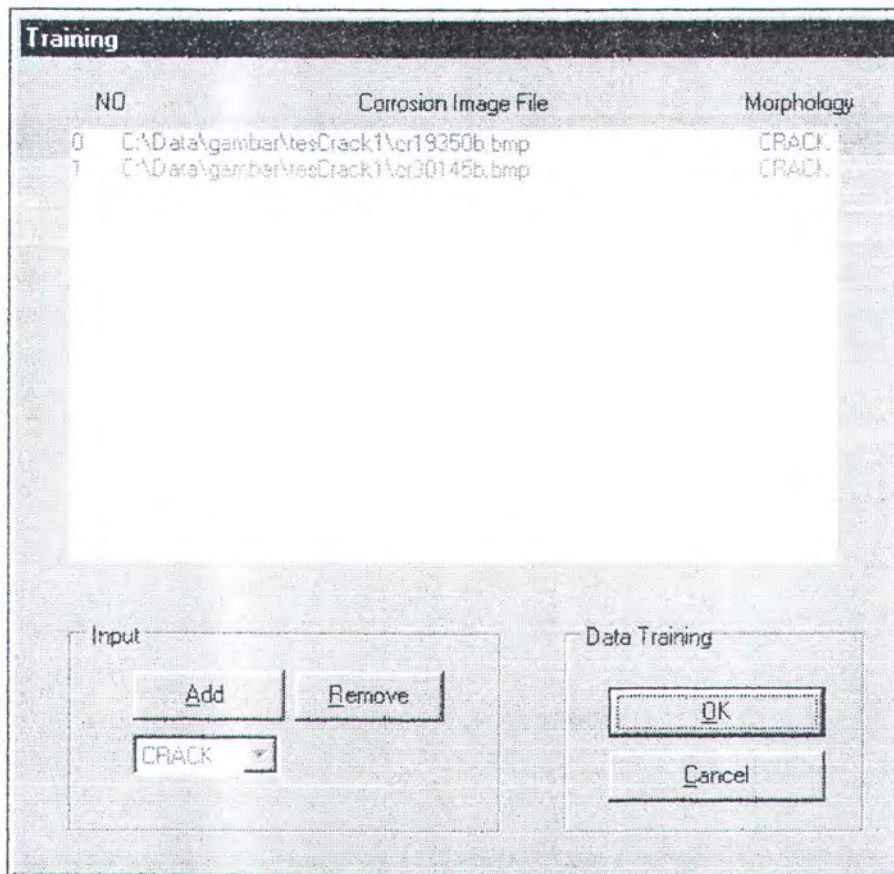
- Pada bagian 'Data pelatihan', pilih satu jenis data masukan yang dipakai, file teks atau file citra.
- Jika data masukan berupa file teks, maka pada bagian 'Data pelatihan-File teks', masukkan file data yang berisi nilai citra (energi dan/atau orian). Jika data masukan berupa file citra, maka pada bagian 'Data pelatihan-File citra', masukkan nama file teks (tanpa ekstensi) yang akan digunakan untuk menyimpan nilai citra yang dihasilkan dari proses pelatihan. File teks ini bisa digunakan untuk data masukan berikutnya untuk jenis masukan berupa file teks.

Perangkat lunak akan secara otomatis menambahkan ekstensi '.DAT' untuk nama file tersebut.

Dalam pelatihan maupun klasifikasi, beberapa file citra dapat digunakan secara bersama-sama untuk mempercepat prosesnya. Untuk sistem ini, banyaknya masukan file citra yang digunakan tidak boleh melebihi 15 buah. Form masukan file citra adalah seperti pada gambar Lampiran.3.

- Nilai  $\alpha$  merupakan nilai konstanta pelatihan dengan nilai :  $0 < \alpha < 1$ . Notasi nilai  $\alpha$  tergantung pada sistem komputer yang dipakai, apakah menggunakan titik (.) atau koma (,) untuk penulisan bilangan desimal, misal : 3,5 atau 3.5.

Steps merupakan jumlah iterasi yang dilakukan selama pelatihan. Nilai steps harus lebih dari 0 dan bilangan bulat. Pada perangkat lunak ini, steps menggunakan tipe *word*, sehingga nilai steps tidak boleh melebihi 65535.

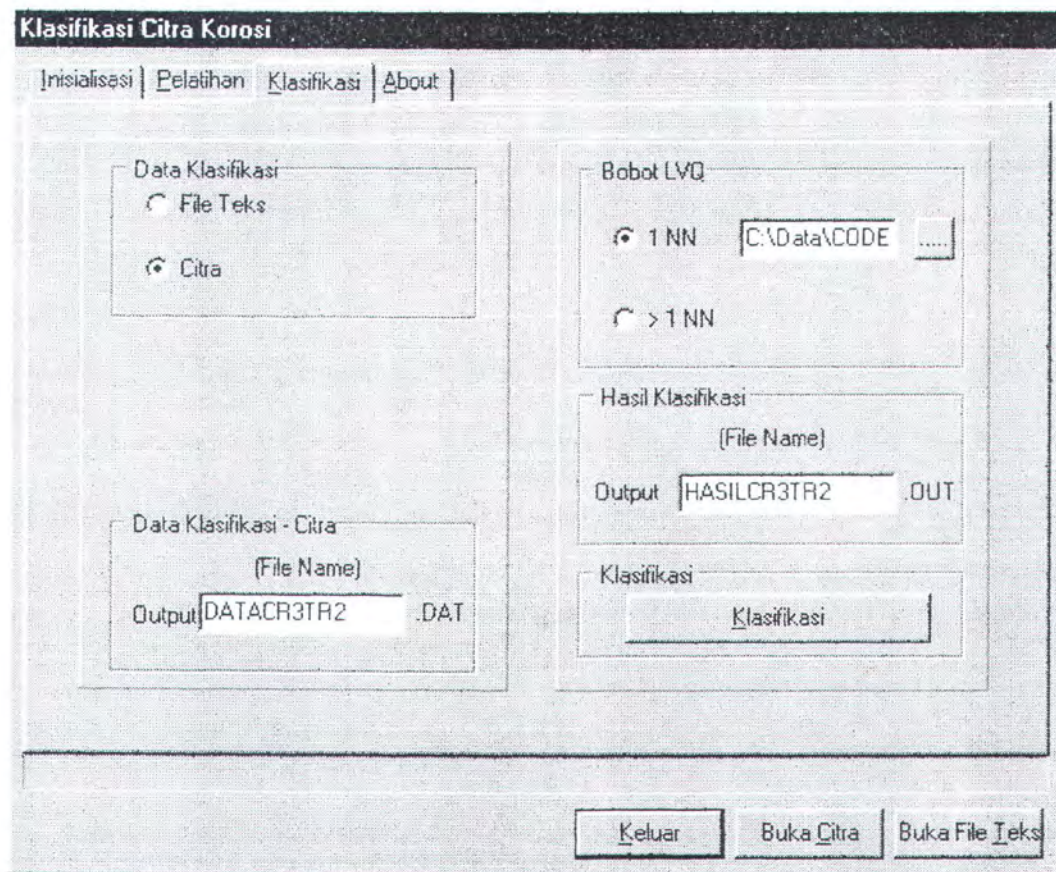


Gambar Lampiran.3 Antarmuka masukan citra

- Pada bagian 'Bobot LVQ', masukkan file teks bobot yang akan dilatih pada subbagian 'input' dan file teks hasil pelatihan pada subbagian 'output'. File teks untuk 'Input' harus berekstensi '.CDBK'. Perangkat lunak ini akan langsung menambahkan ekstensi '.CDBK' pada nama file yang dituliskan pada subbagian 'Output'.
- Pada bagian 'Pelatihan', pilih salah satu versi LVQ yang digunakan, yaitu: LVQ1, LVQ2.1, OLVQ1



### 3. Klasifikasi



Gambar Lampiran.4 Antarmuka proses klasifikasi

Seperti halnya proses pelatihan, proses klasifikasi pun menggunakan dua jenis data masukan, file teks dan file citra.

Proses klasifikasi ini dilakukan dengan cara sebagai berikut :

- Tentukan jenis data masukan
- Jika data masukan berupa file teks, maka pada bagian 'Data klasifikasi-File teks', masukkan file data yang berisi nilai citra (energi dan/atau orian). Jika data masukan berupa file citra, maka pada bagian 'Data klasifikasi-File citra', masukkan nama file teks (tanpa ekstensi) yang akan digunakan untuk menyimpan nilai citra yang dihasilkan dari proses

klasifikasi. File teks ini bisa digunakan untuk data masukan berikutnya untuk jenis masukan berupa file teks. Form masukan citra seperti halnya yang digunakan pada proses pelatihan, dinyatakan dalam gambar Lampiran.3.

Perangkat lunak akan secara otomatis menambahkan ekstensi '.DAT' untuk nama file tersebut.

- Pada bagian 'Bobot LVQ', terdapat dua pilihan bobot LVQ yang digunakan, yaitu '1 NN' jika digunakan satu file bobot LVQ, dan '>1NN' jika banyaknya file bobot yang digunakan lebih dari satu dan kurang dari atau sama dengan tiga.

Jika file masukan berupa file teks dan digunakan '>1NN', maka file teks bobot tersebut harus mempunyai dimensi sama, sedangkan jika file masukan berupa file citra, dimensi bobot bisa berbeda.

- Pada bagian 'Hasil klasifikasi' masukkan nama file yang digunakan untuk menyimpan hasil proses klasifikasi. Perangkat lunak akan langsung menambahkan ekstensi '.OUT' pada file tersebut